NextGen IT
Developer Curriculum Path

(Contract: NextGen IT Assessment contract #35274)

May 9, 2013

Prepared By:

Science Applications International Corporation

Acknowledgements

SAIC would like to thank the State of Tennessee Office for Information Resources for the opportunity to support the State NextGen IT Transformation. After reviewing the Knowledge, Skills, and Abilities (KSAs), Developer tasks, feedback provided by the SMEs; and in consideration of industry best practices, and SAIC experience internally and with other clients, SAIC submits this Developer Curriculum Path (CP).

We also would like to acknowledge and thank the personnel who participated as Subject Matter Experts, and those who provided leadership and support to the project.

Curriculum Development Participants

Pat Buchanan	*Trena Maynard*
Pamela Cobb	*Mary McIntire*
Shannon DeYoung	*Donna Odom*
Kelly Gooch	*Tim Sanford*
Vicky Hutchings	*LaShawn Shaw*
Venkatesh Kondragunta	*Greg Wilkins*
Sheila Mann	*David Wooten*
Tom Sarro	*Franc Stratton*
Liz Moore	*Cliff Morrison*
Khena Kara	

TABLE OF CONTENTS

DEVELOPER CURRICULUM PATH INTRODUCTION

SAIC is pleased to provide this Developer CP document for review by OIR and the selected SMEs in order to facilitate further discussion about the training and labs that would support the NextGen IT objective of improved IT deliverables.

This Curriculum Path for the Software Developer (SD) job classification represents the knowledge, skills, and abilities (KSAs) and tasks associated with the new State of TN job classifications. The CP also reflects input, suggestions, and feedback from State IT Subject Matter Experts (SMEs), and in consideration of IT Industry Best Practices. The following is a summary of the themes that came out of the SME workshop.

- Many people working in IT today did not start out to be in IT but they moved into IT roles as business evolved toward the use of modern technology. As the State works toward better deliverables it will be important to give personnel the opportunity to fill-in and further develop their foundational KSAs even as they are improving their KSAs with current technologies (.NET, Java).
- The NextGen IT CPs should support the use of industry standard SD processes across the State and establish understanding of when and how to tailor the processes to fit the scope of the deliverables. The use of proven, standard yet adaptable processes will give all IT personnel a common language and approach to SD that will enable them to identify problems, reduce errors, and increase customer satisfaction.
- Establishing a culture of informed and engaged IT project teams will positively impact the outcome of deliverables. Training should promote team communication, collaboration, respect for roles, and awareness of the inter-connectivity of the project tasks. As the Developer SMEs pointed-out, developing code without awareness of the larger scope of the project, or the impact of what you produce, is counter-productive. Some described writing code and never knowing if what they did was useful, or effective, or actually used at all. Communication within the IT team and with the customer and end-users increases the likelihood that problems can be identified earlier and addressed prior to production and that the final deliverable will be a better solution.
- Training must improve IT personnel ‘soft’ inter-disciplinary skills and abilities such as communication, critical thinking, decision-making, and active listening. Without these skills and abilities, the IT professional is limited in their ability to effectively use technical knowledge and skill.
- Ultimately, Software Developers will be able to use the Curriculum Path to:
    - Remediate and/or reinforce foundational KSAs required for Developers by completing training required in the beginner levels.
    - Develop and demonstrate the knowledge, skills, and abilities required to perform the tasks relevant to their job classification and level.
    - Develop KSAs to prepare to move into a more advanced Developer level.
    - Develop the ‘cross-training’ KSAs that will improve teamwork, communication, understanding, and ultimately improve deliverables.

Suggested CP Format

This Curriculum Path is presented in a chart format. Job Classification levels are displayed in rows along the left-hand side of the chart (Associate, Junior, Intermediate, and Advanced). All of the training and learning activities that make up the curriculum path are organized into one of five Areas of Competency needed to develop well-rounded IT Professionals (Technical, Organizational / Process, Communications, Cognitive Development, and Administration/Management/Leadership). These Areas of Competency are displayed across the top of the chart. The related courses and learning activities are displayed in these columns according to their relevant job classification level and in the recommended order of completion.

Curriculum Path Format

	AREAS OF COMPENTENCY				
	Technical Training	Organizational / Process Training	Communication Training	Practical Reasoning Training	Administration/ Management/ Leadership Training
Associate Level					
Junior Level					
Intermediate Level					
Advanced Level					

Five Areas of Competency to Develop Well-Rounded IT Professionals

The five Areas of Competency were selected based upon the types of knowledge, skills, and abilities identified in the KSAs for each new State of Tennessee Job Classification and are recommended to emphasize the importance of developing competency in inter-disciplinary knowledge, skills, and abilities in tandem with technical competence.

Technical: The technical courses provide the IT systems developer with the knowledge, skills and abilities to perform the tasks required to analyze, design, develop, test, deploy, and maintain applications pursuant to best practices and user requirements.

Organizational / Process: Organizational and Process Oriented courses are interdisciplinary courses that will be provided for all IT professionals in all State Agencies. The goal is to develop a common set of expectations, and a framework for working together on IT projects regardless of which department and what project. Topics include IT Governance, Software Development Processes, and Project Management Processes.

Communication: Communication is an area of competency that was emphasized over and over again by the SMEs from all job classifications and all State Agencies. These courses will include written and verbal technical communication, active listening, verifying understanding, customer service, and team building.

Practical Reasoning: Courses in this area of competency will expand upon the professional's ability to actively and skillfully conceptualize, apply, analyze, synthesize, and/or evaluate information gathered and to use these skills to create more effective, efficient, and appropriate deliverables within their job classification.

Administration/Management/Leadership: Leadership courses will cover the knowledge, skills, and abilities that are necessary to manage work, motivate and support individuals and teams. These courses are most often going to appear at the more advanced levels of job classifications.

Comprehensive Competency Development and Verification

A significant objective of the NextGen IT transformation is to develop well-rounded IT professionals who can work together to deliver high quality, effective, timely technology products and services to State Government and the people of Tennessee. Instructors will use teaching methods that will maximize the State IT Professional's opportunity to practice, apply, and perform during the class so that the student leaves the classroom and returns to their work place fully prepared to use the new learning in their work. The desired training delivery approach will include instruction and demonstration activities, guided exercises and practice activities, and competency verification activities.

The importance of dynamic, interactive competency verification cannot be overstated. Every course regardless of the type (instructor-led, computer-based, facilitated workshops,) will provide students with at least one competency verification activity or 'lab.' These labs will require the participants to apply the knowledge, skills, and abilities gained to execute a course related task.. A written exam may be used as a competency verification activity; however, the desire is for students to use and apply new knowledge, skills, and abilities prior to leaving the class. This may be accomplished through independent assignments on the computer such as writing code or through activities that requires the students to work together to identify the best technical solution, solve a problem, or produce a product.

These competency verification activities are a core strategy of the new training curriculum, aimed at instituting continuous opportunities to verify, build upon, and improve competency. The expectation is for the student to integrate learning from all five Areas of Competency as they progress through the curriculum path. Competency verification can and should include a requirement for students to use KSAs, obtained in previous classes. Students will understand this expectation and instructors will be responsible to ensure that they are working to support this objective through their classroom, lab, and workshop activities.

Final Competency Verification Lab

In order to complete verification of competency for each job classification level, and /or to move into another job classification, it is recommended that IT professionals pass a final competency lab that will be facilitated by working professionals with appropriate skills and subject matter knowledge. This competency lab will require the participants to use knowledge, skills, and abilities from all five Areas of Competence.

DEVELOPER CURRICULUM PATH

The Curriculum Path for Developers is presented in the charts below. To view the Course Description, Learning Objectives, and Outline click the course title.

Associate Software Developer

	Technical Training	Organizational Process Training	Communication Training	Practical Reasoning Training	Administration/ Management/ Leadership Training
Associate Developer	Software Development Orientation	State of TN IT Professional Orientation	Introduction to Technical Communication	Resources to Improve Practical Reasoning	
	Basic Programming Concepts				
	Improve Skills using Microsoft Professional Tools				
	Associate Software Developer Learning Lab				
	Associate Software Developer Competency Lab				

Junior Software Developer

	Technical Training		Organizational Process Training	Communication Training	Practical Reasoning Training	Administration/ Management/ Leadership Training
Junior Developer	Designing to Maximize Customer Satisfaction		Team Roles and Responsibilities	Technical Documentation Skills		
	.NET Basic A. Integrated Development Environment: .NET	Java Path A. Integrated Development Environment: Eclipse	Time Management	Developing a Customer Service State-of-mind		
	B. Object Oriented Programming and Design: C# and VB.NET	B. Object Oriented Programming and Design: Java				
	C. Data Persistence, Database Connectivity, and Entity Modeling: ADO.NET	C. Data Persistence, Database Connectivity, and Entity Modeling: Oracle				
	D. Client Development: • Thin Client Course: ASP.NET or • Thick Client Course: Visual Studio .NET	D. Client Development: • Thin Client Course: JSF or • Thick Client Course: Eclipse				
	System Analysis Basics					
	Junior Software Developer Learning Lab					
	Junior Software Developer Competency Lab					

Intermediate Software Developer

	Technical Training		Organizational Process Training	Communication Training	Practical Reasoning Training	Administration/ Management/ Leadership Training
Intermediate Developer	Software Design		TN Standard Software Development Process Foundations	Team Dynamics		
	Object Oriented Patterns			Improving understanding between technical and non-technical project stakeholders		
	.NET Intermediate A. Application Security: .NET	Java Intermediate A. Application Security: Java EE 6				
	Application Standards, Best Practices, and Performance Optimization					
	B. Reporting/Logging Technologies: SQL Server	B. Reporting/ Logging Technologies: Jasper				
	System Architecture					
	Intermediate Software Developer Learning Lab					
	Intermediate Software Developer Competency Lab					

Advanced Software Developer

	Technical Training		Organizational Process Training	Communication Training	Practical Reasoning Training	Administration/ Management/ Leadership Training
Advanced Developer	.NET Advanced A. Inter-Application Architecture: WCF	Java Advanced A. Data Exchange and Configuration Support: Java	Project Management Basics	Technical Communications for Leads/Managers	Collaborative Troubleshooting / Problem Solving / Decision Making	Team Building
	B. Workflow Modeling: Windows Workflow Foundation	B. Workflow Modeling: IMIXS		Advanced Team Dynamics		
	Options: • Advanced Programming: C# and VB.NET	Options: • Advanced Programming: Java				
	Introduction to MS Office Integration					
	• Project Lifecycle Coordination for Advanced Software Developers: TFS	• Project Lifecycle Coordination for Advanced Software Developers: Maven				
		• Presentation Skills for Java Programmers				
	Highly Interactive Web Clients					
	Lead Developer Advanced Topics					
	Special Topics					
	Advanced Software Developer Learning Lab					
	Advanced Software Developer Competency Lab					

TECHNICAL COURSE DESCRIPTIONS

The technical courses provide the IT systems developer with the knowledge, skills and abilities to perform the tasks required to analyze, design, develop, test, deploy, and maintain applications pursuant to best practices and user requirements.

CROSS PLATFORM COURSES

Software Development Orientation

This class has a wealth of knowledge that aspiring programmers need to master. This course consists of topics such as: hardware and software components of a computer system, the CPU, registers, operating systems, processors, number systems, data type formats, conversions among number bases, logic and reasoning, flow charting, pseudo code, control structures, differentiation among programming languages, from machine language to assemblers and compilers, data structures, libraries, web programming, and a host of other topics. After learning the fundamentals, students will be given a chance to show their knowledge by writing some simple to intermediate programs in the Java programming language.

Duration: 2 to 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- List the major components of a computer system
- Differentiate between various kinds of computer storage
- Distinguish between system software and application software
- List the steps in the instruction/execution cycle
- Differentiate between various kinds of programming language translators
- Differentiate between various kinds of programming paradigms
- Convert from and to the following bases: binary, octal, decimal, and hexadecimal
- Demonstrate how negative numbers are stored in memory
- Demonstrate how various data types are stored in memory
- State the difference between a program and a process
- Use various graphical techniques, such as flow charting and pseudo code, to specify the logic of a program
- Read BNF notation that describes correct grammar for a programming language
- Compose examples for the primary control flow structures for a programming language
- Compose, compile and run a simple Java application
- State the meaning of the various elements of a simple Java program

Course Outline

- Introduction
  - What is a Computer System?
  - Input Units
  - Output Units
  - Memory
  - Software
  - Components of the CPU
  - Instruction Execution Cycle
  - Cache
  - Stack
  - Functions of an Operating System

- Programming Languages
  - Introduction
  - Machine Language
  - Assembly Language
  - Compilers
  - Backus-Naur Form Notation (BNF)
  - Procedural vs. Object Oriented
  - Scripting vs. Non-Scripting
  - Markup Language
  - Relational Language
  - Choosing a Programming Language
- The Programming Cycle
  - The Software Development Cycle
  - The Programming Cycle - Overview
  - Edit
  - Compile
  - Execute
- Number Systems and Data Types
  - Number Bases
  - Base 10
  - Base 2
  - Base 8 and Base 16
  - Conversion Between Number Bases
  - Data Types
  - Negative Numbers
  - Complement Arithmetic
  - Floating-Point Values
  - String/Character Data
- Programming Skills
  - Spoken Languages vs. Programming Languages
  - Problem Solving
  - Mathematics – Order of Operations
  - Flow Charting
  - Pseudo Code
- Java Introduction
  - What is Java?
  - History
  - Versioning
  - The Java Virtual Machine
  - Writing a Java Program
  - Packages
  - Simple Java Programs
- Java Language Components
  - Primitive Data Types
  - Comments
  - The for Statement
  - The if Statement
  - The while and do while Statements

  - The switch Statement
  - The break Statement
  - The continue Statement
  - The try/catch Statement
  - Operators
  - Casts and Conversions
  - Keywords
- User Interfaces
  - Interfaces
  - Character Based - Command Line
  - Character Based - Interactive
  - Graphical User Interfaces
  - Client/Server Computing
  - Web Based Applications

Lab Exercises: Using command line execution, the attendee will create and run a simple "Hello World" program in Java. Given a simple business process, the attendee will pseudo code a solution and further code the solution in Java using the language components presented in the class.

Basic Programming Concepts

This course is a basic introduction to programming concepts including the software development life cycle, common object-oriented programming topics, data structures, database design, testing techniques, networks, and internet protocols. Common technical terminology and acronyms will be presented.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Explain the processes in the SDLC
- Integrate basic programming concepts into development
- Identify when to use object-oriented programming and the benefits associated with it
- Use data structures and algorithms effectively in program design and development
- Design a simple database in third normal form
- Recognize different internet protocols, services, and environments

Course Outline

- Software Development Lifecycle
  - The SDLC Process Model
  - Development Roles
  - Analysis and Design Phases
    - Requirements Analysis
    - Introduction to the Unified Modeling Language (UML)
    - Introduction to Use Cases
  - Implementation, Testing, and Documenting
    - Testing Techniques
  - Deployment and Maintenance
- Programming Concepts
  - Program Flow
  - Syntax, Operators, Variables and Statements
  - Conditional Branching and Iterative statements; if, for, while
  - Functions, Nested Conditionals and Switch Statements

  - Logical Operators and Arrays
  - Multi-dimensional Arrays
  - Debugging techniques
  - Identifying Syntax Errors
  - Identifying Logic Errors
- Object-Oriented Programming
  - Introduction to the Object-Oriented Programming Paradigm
  - Most appropriate applications of Object-Oriented Programming
  - Cost Benefits, Improved Software Reliability and Maintainability
  - Criteria for language selection amongst .NET and Java
  - Object Oriented Libraries
  - Methodologies for Object-Oriented Development
- Data Structures and Algorithms
  - Analysis of algorithms
  - Lists
  - Trees, Binary Search Trees, Multi-Way Search Trees
  - Dictionaries, Hash Tables
  - Graphs, Graph Traversals, Graph Algorithms
  - Sorting
- Introduction to Database Design
  - Relational Database Model
  - Entities and Tables
  - Attributes and Columns
  - Data Definition Language (DDL)
  - Data Manipulation Language (DML)
  - Data Query Language (DQL)
  - Rudimentary SQL Commands
  - Keys
  - Normalization and Normal Forms
  - Entity Relationship (ER) Modeling Diagrams
  - Indices
  - Client/Server Model
  - Databases with Applications/Databases with Web
- Internet Protocols
  - Introduction to HTTP, HTTPS, FTP, and SFTP
  - Introduction to TCP/IP
  - IP Addressing
  - Network TCP/IP Protocols
  - Basic TCP/IP Services
  - Domain Name System (DNS)
  - OSI, SSH, SMTP, POP, Telnet
  - SSL, TLS
  - Securing TCP/IP Environments
  - Routing in the IP Environment
  - TCP/IP, NETBIOS, and WINS

Lab Exercises: The exercises for this course will be varied due to the range of topics covered: Group discussion of the SDLC with opportunities to develop simple UML and Use Case diagrams; Pseudo Code with the Programming Concepts and Data Structures covered; Debugging exercise; Object-Oriented development; Simple Database Design; and Internet Protocols Troubleshooting.

Improve Skills using Microsoft Professional Tools

This is a set of Free Microsoft CBTs that will assist personnel 'just-in-time' with their user of Microsoft Professional Office products including Includes Word, Excel, PowerPoint, Outlook, Access, Lync, OneNote, Project, SharePoint, Visio, Office 365 for business, Communicator, and Live Meeting.

Duration: varied

Audience: All IT professionals

Learning Objectives

- The objective is to increase general knowledge and skills with these tools when needed.

Course Outline

- Defined by Microsoft as presented at: http://office.microsoft.com/en-us/training-FX101782702.aspx

Associate Software Developer Learning Lab

This lab will prepare the participant for the Competency Lab. Exercises will be provided that will demonstrate the participant's readiness to complete the Competency Lab.

Associate Software Developer Competency Lab

This lab will provide the opportunity for the developer to demonstrate their ability in the skills necessary to advance to the next job classification level.

Designing to Maximize Customer Satisfaction

This course is an introduction to visual software design concepts including user interface design considerations, creating applications that meet ADA standards, and an introduction to style sheets. It prepares the IT software developer to provide quality services for the customer/end-user. Other topics designed to enhance customer satisfaction include: identifying stakeholders, managing expectations, user acceptance testing, communication, and post-implementation evaluation.

Duration: 2 or 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Identify the stakeholder/customer
- Manage expectations
- Create a prototype for a given application
- Understand the importance of user acceptance testing
- Create a user interface that meets ADA minimum standards
- Create a user interface that operates properly regardless of platform
- Create a user interface that meets user requirements
- Create a user interface that has proper navigational controls
- Create a user interface for the desktop environment
- Create a user interface for the internet
- Identify good vs. poor user interface design
- Create a style sheet
- Perform a post-implementation evaluation

Course Outline

- Who is the Customer?
  - Stakeholders/Customers
- Managing Expectations

    - Communication with the Stakeholder
    - Developing Prototypes for Stakeholder Approval
        - Why prototype?
        - Types of prototypes
        - How to create a prototype
- User Acceptance Testing
    - What is UAT?
    - When should it be done?
    - What should be done with the results?
- Introducing human-computer interaction
    - What is a user interface?
    - Attributes of a good interface
    - Leveraging our cognitive skills
    - Leveraging our perceptual skills
    - ADA Standards
    - Non-visual Factors
- Introducing user-centered design
    - User-centered design
    - Design iteration
    - Analysis phase
    - Design phase
    - Verification phase
- User interface design principles
    - Attributes of a good interface
    - Map to the user's mental model
    - Be consistent
    - Provide control
    - Provide feedback
    - Recover gracefully
    - Provide flexibility
- Navigation
    - How to create a task flow diagram
    - Five navigational structures
        - Sequential navigation
        - Hierarchical navigation
        - Star navigation
        - Grid navigation
        - Network navigation
        - Hybrid navigation
- Application interfaces
    - Primary window
    - Secondary windows
    - Menus
    - Controls
    - Toolbars
    - Status bar
    - Input: General
    - Input: Mouse Input
    - Input: Keyboard
    - Interaction: Selection operations
    - Interaction: Editing operations
    - Interaction: Transfer operations
    - Windows user assistance and help

    - Writing for desktop applications
    - Platform specifics
- Web interfaces
    - Client-server explained
    - Client-server performance issues
    - Introduction to Style Sheets
    - Web visual design
    - Links
    - Platform or browser
- Post-Implementation Evaluation
    - What information should be gathered?
    - Implementing Changes
    - Documenting for Future Projects

Lab Exercises: Attendees will be presented with examples of both good and poor GUIs without distinction. They will determine what is appropriate and effective and what is poor design and ineffective. Given a simple project, attendees will create a prototype of the interface and use a style sheet to modify its appearance.

System Analysis Basics

This course is an introduction to analysis and evaluation of systems operations. Topics include: technology design, business requirements, business processes and quality control. Using best practice standards, the learner will evaluate various portions of the SDLC deliverables for appropriate content, efficiency, and effectiveness.

Duration: One Day

Learning Objectives

Upon completion of this course, attendees will be able to:

- Recognize effective hardware design
- Recognize effective business requirements and know what to do when they are lacking
- Effect change in inferior business processes
- Plan for quality control through UAT and other methods

Course Outline

- Technology Design Analysis
    - Determining Delivery Method
    - Assessing Hardware Requirements
    - Evaluating Hardware Design Alternatives
    - Cost/Benefit Analysis
    - System Testing Techniques
- Business Requirements Analysis
    - Types of Requirements Errors
    - Actions to Take When Requirements are Lacking
    - Characteristics of Well-Written Requirements
    - Requirements and the Development Lifecycle
    - Identifying Constraints and Benefits
- Business Processes Analysis
    - Determining Best Practice
    - Obtaining Feedback for Improvement
    - Developing an Improvement Plan
    - Gaining Approval for Change

- Quality Control Analysis
  - Planning for Quality
  - Acceptance Sampling (User Acceptance Testing)
  - Quality Improvement Methods and Tools

Lab Exercise: The exercises for this course will be group centered case scenarios in each of the areas covered. Attendees will identify examples given as having good or poor qualities in the identified area. Groups will also create a sample improvement plan for a simple process; develop a simple cost/benefit analysis for a small project; and develop several well-written business requirements.

Junior Software Developer Learning Lab

This lab will prepare the participant for the Competency Lab. Exercises will be provided that will demonstrate the participant's readiness to complete the Competency Lab.

Junior Software Developer Competency Lab

This lab will provide the opportunity for the developer to demonstrate their ability in the skills necessary to advance to the next job classification level.

Software Design

This course covers the basics of functional software design. Utilizing the skills obtained in the Introduction to Software Design course, the learner will further explore the issues related to functional user interface design, prototyping. Different software design methodologies such as Agile, Scrum, and Waterfall will be discussed.

Duration: 2 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Integrate UML, Agile, Test Driven Development (TDD) and Object Oriented Analysis and Design (OOAD) into the SDLC when appropriate
- Apply design patterns in business applications
- Articulate the differences in the design methodologies Agile, Scrum, and Waterfall

Course Outline

- Test Driver Development
- Applied OO
- UML Essentials
- Agile Development
- Commonality and Variance
- Delegation
- Refactoring
- Using Abstraction
- Scrum
- Waterfall
- How Does Design Affect Process?

Lab Exercises: Case scenarios will be presented as group exercises to design simple business applications using the design topics presented.

Object-Oriented Patterns

This course includes an introduction to the following patterns: distributed application, interface, responsibility, construction, operation, extension and application. The learner will use these patterns to create a Windows form application for a simple business process.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Use these patterns to create a Windows form application

Course Outline

- Introduction and History of Distributed Application Patterns
- Interface Patterns (Adapter, Façade, Composite , Bridge)
- Responsibility Patterns (Singleton , Observer, Mediator, Proxy, Chain of Responsibility, Flyweight)
- Construction Patterns (Builder, Factory Method, Abstract Factory, Prototype, Memento)
- Operation Patterns (Template Method, State, Strategy, Command, Interpreter)
- Extension Patterns (Introducing Extensions, Decorator, Iterator, Visitor)
- Application Patterns: Model-View-Controller, Model-View-Presenter, Model-View-View-Controller

Lab Exercises: Labs are needed that demonstrate the key concepts presented above.

Application Standards, Best Practices and Performance Optimization

This course is designed to enable the developer to enhance application performance through the utilization of standards and best practices. Topics include: identifying and evaluating performance issues, performing architecture and design reviews, choosing a deployment topology, designing for required performance and scalability, passing data across tiers, choosing between service and object orientation, improving managed code performance, managing memory efficiently, using multithreading in .NET applications, cleaning up resources, handling exceptions, working with strings efficiently, improving data access performance, caching data and other objects, reliability, throughput, and measuring performance.

Duration: 5 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Identify and evaluate performance issues
- Design for reliability and throughput
- Perform architecture and design reviews
- Choose a deployment topology
- Design for required performance and scalability
- Pass data across tiers
- Choose between service and object orientation
- Improve managed code performance
- Manage memory efficiently
- Use multithreading in .NET applications
- Clean up resources
- Handle exceptions
- Work with strings efficiently
- Improve data access performance
- Cache data and other objects

Course Outline

- Balancing performance with quality-of-service (QoS) requirements: Balancing application performance requirements with other QoS attributes such as security and maintainability. This section presents Microsoft design guidelines for application performance.
- Identifying and evaluating performance issues: Using performance modeling early in the design process to help evaluate design decisions against the objectives before committing time and resources. This section includes identifying performance objectives, performance modeling, team workload, and performance metrics such as maximum execution time and resource utilization (e.g., CPU, memory, disk I/O, and network I/O).
- Performing architecture and design reviews: Review the design of an application in relation to the targeted deployment environment, any constraints that might be imposed, and the defined performance goals. Use the categories that are defined by the performance and scalability frame promoted by this guide to help partition the analysis of an application and to analyze the approach taken for each area. The categories represent key areas that frequently affect application performance and scalability. Use the categories to organize and prioritize areas for review.
- Choosing a deployment topology: Identify the requirements and constraints that exist between the application and infrastructure. This precedes choosing appropriate architectures and helps resolve conflicts between application and infrastructure architecture early in the software development process. This section covers a layered design that includes presentation, business, and data access logic that makes it easier for the developer to scale an application and improve maintainability.
- Designing for required performance and scalability: Using tried and tested design principles with a focus on the critical areas where the correct approach is essential and where mistakes are often made. This section should show how to use the categories o help organize and prioritize performance issues. Categories include data structures and algorithms, communication, concurrency, resource management, coupling and cohesion, and caching and state management.
- Passing data across the tiers: Prioritize performance, maintenance, and ease of development when you select an approach. Custom classes allow a developer to implement efficient serialization. This section should also cover using structures if you can to avoid implementing serialization or XML for interoperability and flexibility with precautions since XML is verbose and can require considerable parsing effort. This section also includes using a DataReader object to render data as quickly as possible, but not passing DataReader objects between layers because they require an open connection. This section should discuss DataSet options for flexibility and caching data across requests, but DataSet and Typed DataSet objects are expensive to create and serialize. This section finally covers passing objects, generics, and lists between layers.
- Service Orientation vs. Object Orientation. When designing distributed applications, choosing among services, object-orientation, or a hybrid of both can be a difficult choice. This section summarizes the key differences between object orientation and service orientation.
- Improving Managed Code Performance: Conducting performance reviews of managed code. This section should cover performance issues when designing efficient types and thread safe classes, properties, and inheritance.
- Memory Management: Writing code to help the garbage collector do its job efficiently. This section should also include minimizing hidden allocations, avoiding the promotion of short-lived objects, pre-allocating memory, chunking memory, and forcing garbage collections.
- Multi-Threading in .NET Applications: Minimizing thread creation and using the self-tuning thread pool for multithreaded work. This section should also cover avoiding the creation threads on a per-request basis and the use Thread.Abort or Thread.Suspend. Make sure that you appropriately tune the thread pool for ASP.NET applications and for Web services. For more information, see "How to tune the ASP.NET thread pool" later in this document.
- Managing Resources: Releasing resources as soon as an application is finished with them. This section covers using “finally” blocks or the C# “using” statement to make sure that resources are released even if an exception occurs. It also discusses calling Dispose (or Close) on any disposable object that implements the IDisposable interface. The section should also include using finalizers on classes that hold on to unmanaged resources across client calls. It should also include using the Dispose pattern to help ensure that you implement Dispose functionality and finalizers (if they are required) correctly and efficiently.

- Handling Exceptions: Writing code that avoids unnecessary exceptions. This section should cover using finally blocks to guarantee resources are cleaned up when exceptions occur (e.g., closing database connections in a finally block). It also covers exception handling best practices for layered or n-tiered applications and services.
- Efficient use of Strings: Using excessive string concatenation results in many unnecessary allocations that create extra work for the garbage collector. This section covers the use of the "StringBuilder" when creating complex string manipulations and concatenating strings multiple times.
- Improving Data Access Performance: This section covers improving data access performance by minimizing processing on the server and at the client and minimizing the amount of data passed over the network. This section should also include using database connection pooling to share connections across requests and keeping transactions as short as possible to minimize lock durations and to improve concurrency. The topic should cover paging records for grids in web and Windows applications.
- Caching data and other objects: Caching ASP.NET data by using the Cache API, by using output caching, or by using partial page fragment caching. This section should cover the consideration necessary for an appropriate caching policy that identifies the data for caching, the place for caching, and updating the cache.
- Measuring performance: Measuring performance early in the application design phase using tools such as System Monitor, network monitoring tools such as Netmon, profiling tools such as CLR Profiler, SQL Profiler, SQL Query Analyzer, and application instrumentation to collect metrics for measuring, This section also covers Quality Assurance measurements such as load testing and other performance testing.

Lab Exercises: Labs and demonstrations are needed that demonstrate the key concepts presented above.

System Architecture

This course addresses hardware architecture considerations that affect design and development including the hardware for the chosen delivery method (application, database, web), network limitations, network security, and firewall rules.

Duration: One Day

Learning Objectives

Upon completion of this course, attendees will be able to:

- Identify appropriate system architecture for a given application
- Recognize network limitations
- Develop network security protocols
- Evaluation basic firewall technology, features and hardware models

Course Outline

- Choosing Appropriate Hardware for Delivery Method
  - Database Driven
  - Desktop Application
  - Web Delivery
- Network Limitations
- Network Security
- Firewalls

Lab Exercises: Case scenarios will be given to attendees in groups to identify and evaluate appropriate architecture choices in the areas presented in the course.

Intermediate Software Developer Learning Lab

This lab will prepare the participant for the Competency Lab. Exercises will be provided that will demonstrate the participant's readiness to complete the Competency Lab.

Intermediate Software Developer Competency Lab

This lab will provide the opportunity for the developer to demonstrate their ability to design exemplary application modules and to analyze the quality of a design. Model a Census Data Collection Application.

Introduction to MS Office Integration

This course includes topics related to MS Office programming and MS Office solutions. The learner will be introduced to MS Office programming in .NET. Further content includes VSTO extensions, using Windows forms, working with Smart Tags, and data programming. The course concludes with server data usage, security, and VSTO deployment.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Develop an application using VSTO

Course Outline

- Introduction to Office Programming: Office Object Models, Properties, Methods and Events, and The Significance of Primary Interop Assemblies.
- Introduction to Office Solutions: Office Automation Executables, Office Add-Ins, and Code Behind a Document.
- Office Programming in .NET: Programming Excel, Working with Excel Events, Working with Excel Objects, Programming Word, Working with Word Events, Working with Word Objects, Programming Outlook, Working with Outlook Events, Working with Outlook Objects.
- The VSTO Programming Model: VSTO Extensions to Word and Excel Objects, Dynamic Controls, and VSTO Extensions to Word and Excel Object Models.
- Using Windows Forms in VSTO: Adding Windows Forms Controls to Your Document, Writing Code Behind a Control, The Windows Forms Control Hosting Architecture, Properties Merged from OLEObject or OLEControl, Adding Controls at Runtime, Working with an ActionsPane, Working with the ActionsPane Control.
- Working with Smart Tags in VSTO: Creating Document-Level Smart Tags with VSTO, and Creating Application-Level Smart Tags.
- VSTO Data Programming: Creating a Data-Bound Customized Spreadsheet with VSTO, Creating a Data-Bound Customized Word Document with VSTO, Datasets, Adapters, and Data Sources.
- Server Data Scenarios: Populating a Document with Data on the Server, Using ServerDocument and ASP.NET, The ServerDocument Object Model.
- .NET Code Security: Code Access Security Versus Role-Based Security, Code Access Security in .NET, Code Origination Location, Strong Names, Publisher Certificates, Trusting the Document, and Deploying Policy to User Machines.
- Deployment: VSTO Prerequisites, Deploying to an Intranet Shared Directory or Web Site, Editing Manifests, and Creating Setup Packages.

Lab Exercises: Three Labs are needed that illustrate the key concepts presented above in a scenario that illustrates the non-functional, technical best practices for application performance. The three labs should be about 1 or 2 hours at the end of each day and divide topic numbers 1 through 10 appropriately. Object-orientation should always be included in all labs.

Highly Interactive Web Clients

This course includes the fundamentals of AJAX, JavaScript, JSON, and AJAX controls. Client side development, application services, and forms authentication are presented.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Develop web applications utilizing jQuery
- Develop web applications utilizing AJAX
- Develop web applications utilizing Object-Oriented Programming with JavaScript

Course Outline

- Introduction to AJAX
- History of Web Applications
- The User Experience
- JavaScript Fundamentals
- Overview of JavaScript Object-Oriented Programming with JSON
- The XMLHttpRequest object
- Overview ASP.NET AJAX
- Typical ASP.NET Application
- Introduction to jQuery
- jQuery Selectors
- jQuery Events
- jQuery Event Handler Methods
- jQuery Special Effects
- Simple Page Styling with jQuery and CSS
- jQuery Callbacks
- jQuery Page Method
- jQuery Data Method
- Advanced Dirty Flag with jQuery
- jQueryUI Controls
- Designing the Server API for Remote Calls
- AJAX Call to Web Service
- Consuming Web Services with Soap/XML
- Consuming Web Services with JSON Data
- Page Method Calls to WCF Services

Lab Exercises: Labs are needed that demonstrate the key concepts presented above in a scenario that illustrates simple steps in an SDLC (ALM): requirements, UML and functional and non-functional requirements.

Lead Developer Advanced Topics

This is an advanced course covering the following topics: database design, internet protocols, XML Web Service, error handling and troubleshooting.

Duration: 4 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Design a simple database in third normal form
- Establish a secure environment using firewall settings and SSL certificates
- Utilize XML web services to create more robust web applications
- Perform code reviews
- Establish advanced error handling and troubleshooting protocols

Course Outline

- Database Design
    - Advanced Relational Database Design

  - Normalization
  - Distributed Data Management
  - Keys
  - Advanced SQL Queries
- Internet Protocols
  - Advanced Firewall Setup
  - SMTP
  - SSL and Certificates
- Web Services as a Distributed Resource
- Distributed Application Processing
  - Load Balancing
  - System Interfaces
  - Java Messaging System
  - Advanced Enterprise Java Beans
- Code Reviews
  - Best Practices
  - Reusability
  - Maintainability
  - Scalability
  - Reliability
  - Usability
  - Performance
  - Repeatability
- Advanced Error Handling and Troubleshooting
- Project Coordination Skills
  - Project Knowledge Base
  - Robust Code Management
  - Artifact Profiles
  - Environment Configuration and Promotion
  - Build Procedure
  - PowerShell Automation

Lab Exercises: The labs for this course will include group collaboration with practice in each of the topics covered.

Special Topics

This is an advanced course covering the latest emerging technologies.

Advanced Software Developer Learning Lab

This lab will prepare the participant for the Competency Lab. Exercises will be provided that will demonstrate the participant's readiness to complete the Competency Lab.

Advanced Software Developer Competency Lab

This lab will provide the opportunity for the lead developer to demonstrate their ability to lead a project team, manage a project, and perform all the skills necessary to advance beyond lead developer. Build a highly scalable eCommerce Application and Perform Volume Testing.

.NET COURSES

Integrated Development Environment: .NET

This introductory course includes an overview of object-oriented programming, the .NET Integrated Development Environment (IDE) and the Software Development Life Cycle (SDLC). Various application features are presented to provide the learner with the skills necessary to navigate within the application user interface and to create a simple data entry form based upon functional requirements.

Duration: 2 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Navigate within the .NET IDE
- Utilize the topics covered to create a Windows Form application

Course Outline

- Visual Studio Integrated Development Environment (IDE) Overview
- Visual Studio Menu Bar and Toolbar Features
- Visual Studio Windows Overview
- Visual Studio Solution Explorer Features, Buttons on the Solution Explorer Window
- Visual Studio Toolbox
- Visual Studio Properties Settings
- Visual Studio Class View Features
- Visual Studio Object Browser Features
- Visual Studio Task List Features
- Visual Studio Document Windows
- Visual Studio Help
- Visual Studio Code Editor (IntelliSense, Split the Editing Window, Find/Replace, Block Comment/UnComment)
- Visual Studio Creating New Projects and Solutions
- Brief Introduction to the Software Development Process (A Software Development Life Cycle that Includes Interviews, Business Requirements, Functional Requirements, Design, Implementation, Testing, Deployment)
- Windows Form Project Creation from SDLC Documentation
- Creating a Windows Form (Drag and Drop)
- Handling Form Events
- Project Versions
- Project Files
- Single and Multiple Project Solutions
- Form Controls (Setting Tab Order, Set the Font Property, Manipulating Groups of Controls, Formatting Controls)
- Control Events
- Microsoft .NET Namespaces
- Custom Namespaces

Lab Exercise: A simple program lab that includes the SDLC steps above to create a Windows Form application implementing a very simple online phone call log business process.

Object Oriented Programming and Design: C# and VB.NET

The main topics of this course include .NET object fundamentals, object orientation and design considerations. The fundamentals from the previous course are utilized to create a more complex solution. Exception handling is introduced in this course.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Utilize the topics covered to create a more complex Object-Oriented .NET application

Course Outline

Overview of Visual Basic .NET and C# Object-Oriented Programming in .NET Framework

- .NET Object Fundamentals
- Creating and Compiling the Component
- Component Namespaces
- Using a Component
- Application Domains
- Contexts
- Assemblies
- Intermediate Language
- The Global Assembly Cache
- System Namespaces
- Class Anatomy
- Member Variables
- Properties
- Methods
- Access Modifiers
- Passing Parameters
- Value Types
- Creation and Destruction
- Delegates and Events
- Design Considerations

Lab Exercise: A lab is needed that demonstrates the key concepts presented above in a scenario that illustrates simple steps in an SDLC: an interview, business requirements, and functional requirements.

Object-Orientation

- Generalization and Specialization
- Inheritance
- Containment and Data Hiding
- Polymorphism
- Overloading
- Overriding
- Substitution (Liskov)
- Interfacing .NET
- Private Implementation
- Versioning
- Interfaces –vs- Abstract Base Classes
- Interfaces in .NET

Lab Exercise: A lab is needed that demonstrates the key concepts presented above in a scenario that illustrates simple steps in an SDLC: an interview, business requirements, and functional requirements.

Design Considerations

- Exception Handling Objects
- The Basics of Exception Handling
- An Unhandled Exception Handler

- The StackTrace Object
- Resuming Code
- Retrying Code
- Performance Counters
- Object Inspection
- Reflection
- Runtime Type Discovery
- Dynamic Type Loading
- Attributes
- Custom Attributes
- Passing Objects as Parameters
- Streams
- Readers and Writers
- Serialization
- Schema Definition Tool
- Custom Serialization
- NetworkStream

Lab Exercise: A lab is needed that demonstrates the key concepts presented above in a scenario that illustrates simple steps in an SDLC: an interview, business requirements, and functional requirements.

Data Persistence, Database Connectivity, Entity Modeling: ADO.NET with SQL Server

This course serves as an introduction to databases, database design, and database queries. It includes an introduction to SQL Server and Management Studio, an introduction to relational databases, an introduction to XML, an introduction to ADO.NET, and an introduction to Entity Framework. Emphasis is placed on successfully creating database queries using both Query Analyzer and ADO.NET and using this data to fill both web and Windows forms.

Duration: 7 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Create a database query using Query Analyzer
- Create a database query using ADO.NET
- Access data from a database and populate web and Windows forms

Course Outline

Introduction to SQL Server and Management Studio

- Install/Setup SQL Server Express and Management Studio
- How to use Management Studio to Create a Database
- Introduction to Relational Databases, Tables, Primary Keys, Foreign Keys, Indexing, and Constraints
- Writing Database Queries in the Query Analyzer
- Manipulating Data with Queries
- Writing and Executing Simple Select, Insert, Update, and Delete Stored Procedures with Parameters
- Introduction to XML
- Understanding SQL Server Transactions
- Creating a Simple Entity-Relationship Diagram with Management Studio

Lab Exercise: A lab is needed that demonstrates the key concepts presented above in a scenario that illustrates simple steps in an SDLC: an interview, business requirements, and functional requirements.

Introduction to ADO.NET

- Overview of the ADO.NET Objects
- Making Connections
- Executing Commands and Stored Procedures
- Creating and Passing Parameters
- Using Data Readers
- Using Datasets and Data Adapters
- SQL Server Exception Handling
- Select, Insert, and Update Stored Procedures with ADO.NET
- Handling Exceptions specific to Data Access
- Working with Events
- Working with Text and Binary Data
- The ADO.NET Data Access Layer Class
- Windows Forms Application with Data
- Web Forms Application with Data / ASP.NET Applications

Lab Exercise: A lab is needed that demonstrates the key concepts presented above in a scenario that illustrates simple steps in an SDLC: an interview, business requirements, and functional requirements.

Introduction to Entity Framework (EF)

- Overview of the Basic Architectural Model in the EF API and Objects
- Using Entity Framework to give life to models allowing developers to query entities and relationships in the domain model
- Creating and Mapping a Conceptual Model – Understanding Code First
- Accessing and Changing Entity Data
- Entity Framework Data Providers
- Entity Data Model Tools
- Development and Deployment Considerations
- Windows and Web Form Applications

Lab Exercise: A lab is needed that demonstrates the key concepts presented above in a scenario that illustrates simple steps in an SDLC: an interview, business requirements, and functional requirements.

Thin Client Development: ASP.NET

This course includes introductions to ASP.NET web programming, security, designing the user interface, data binding, web forms programming, web services, JavaScript, ASP.NET AJAX and IIS. Using the information obtained in this course, the learner will create a functional, effective and attractive ASP.NET form application with database interaction, data validation and security protocols in place.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Create a functional, effective and attractive ASP.NET form application with database interaction, data validation and security protocols in place

Course Outline

- Introduction to ASP.NET Web Programming with Visual Studio .NET
- Creating Web Forms Applications from Business and Functional Requirements
- Working with Web Objects, Properties, and Events
- Federal and State User Interface Standards
- Handling Data with an ADO.NET Web Application
- Catching and Correcting Errors

- Maintaining Security
- Web Applications in Layers and Assemblies
- Creating Custom Web Controls
- Optimizing Web Applications with Caching
- Formatting Web Application Output
- Providing Help
- Globalizing Web Applications
- Working with Form Objects and Events
- HTML Controls, Properties, and Events
- Server Controls, Properties, and Events
- User Controls, Properties, and Events
- Validating User Input at User and Business Layers
- Formatting with Cascading Style Sheets
- Event Handling in ASP.NET Forms and Components
- Error Handling of Exceptions
- Handling State: Session, Application, and Data Storage
- Unit Testing and Debugging Web Applications
- Introduction to Data Binding in ASP.NET
- Web Forms Programming with Class Layers (User Interface Layer, Business Logic Layer, Data Access Layer)
- Web Configuration and Resource Files
- Introduction to Web Services
- Creating a Simple Web Service
- Introduction to JavaScript and ASP.NET AJAX
- Web Application Deployment
- Introduction to IIS

Lab Exercises: Labs are needed that demonstrate the key concepts presented above in a scenario that illustrates simple steps in an SDLC: an interview, business requirements, functional requirements, design, and programming.

Thick Client Development: Visual Studio .NET

This course includes introductions to Windows Programming with Visual Studio .NET, security, designing the user interface, data binding, Windows forms programming, Windows services, Smart Client applications and Visual Studio Tools for Office. Using the information obtained in this course, the learner will create a functional, effective and attractive Windows form application with database interaction, data validation and security protocols in place.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Create a functional, effective and attractive Windows form application with database interaction, data validation and security protocols in place

Course Outline

- Introduction to Windows Programming with Visual Studio .NET
- Creating Windows Forms Applications from Business and Functional Requirements
- Working with Windows Forms Events and Properties
- Federal and State User Interface Standards
- Handling Data with an ADO.NET Windows Application
- Catching and Correcting Errors
- Maintaining Security
- Windows Applications in Layers and Assemblies
- Creating Custom Controls

- Windows Forms and the Multiple Document Interface
- Formatting Windows Application Output
- Providing Help
- Federal and State Standards and Windows Applications
- Working with Form Objects and Events
- Controls, Properties, and Events
- Server Controls, Properties, and Events
- User Controls, Properties, and Events
- Validating User Input at User and Business Layers
- Formatting Windows Forms and Controls
- Event Handling in ASP.NET Forms and Controls
- Error Handling of Exceptions
- Unit Testing and Debugging Windows Applications
- Introduction to Data Binding in Windows Forms
- Windows Forms Programming With Class Layers (User Interface Layer, Business Logic Layer, Data Access Layer)
- Windows Configuration and Resource Files
- Introduction to Windows Services
- Creating a Simple Windows Service
- Introduction to Smart Client Applications
- Introduction to Visual Studio Tools for Office
- Windows Application Deployment

Lab Exercises: Labs are needed that demonstrate the key concepts presented above in a scenario that illustrates simple steps in an SDLC: an interview, business requirements, and functional requirements.

Application Security: .NET

This is a comprehensive course on system security and data privacy issues. The learner will be able to assess existing infrastructure and applications, identify threats and vulnerabilities and be able to eradicate them. Methods for designing and developing new secure web applications will be presented in detail. Data privacy standards and policies will be discussed. Emphasis will be placed on how to develop applications that are compliant with the existing standards and policies.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Assess existing infrastructure and applications
- Identify threats and vulnerabilities
- Eradicate security threats and vulnerabilities
- Employ methods for securing web applications
- Protect data from piracy and adhere to data privacy standards

Course Outline

- Understanding Basic Threat Terminology: Asset, Threat, Vulnerability, Attack (or Exploit), and Countermeasure.
- Introduction to Threats and Countermeasures: Identifies and illustrates the various threats facing the network, host, and application layers. Using the threat modeling process allows identification of the threats that are relevant to a .NET application. This sets the stage for identifying effective countermeasures. This part includes: Web Application Security Fundamentals, Threats and Countermeasures, Threat Modeling.
- Designing Secure Web Applications: Provides the guidance needed to design Web applications securely. This section identifies the concepts, principles, and techniques that are used during an application design. This part includes: Design Guidelines for Secure Web Applications, and Architecture and Design Review.

- Building Secure Web Applications: Helps apply the secure design practices and principles covered in the previous part to create a solid and secure implementation. This section promotes learning defensive coding techniques that make .NET code and applications resilient to attack. This section also presents an overview of the .NET Framework security landscape so that programmers are aware of the numerous defensive options and tools that are at their disposal. This part includes: .NET Security Fundamentals, Building Secure Assemblies, Code Access Security in Practice, Using Code Access Security with ASP.NET, Building Secure ASP.NET Pages and Controls, Building Secure Serviced Components, Building Secure Web Services, and Building Secure Data Access.
- Securing a Host and Application: Shows how to apply security configuration settings to secure the interrelated network, host, and application levels. Rather than applying security randomly, the programmer should learn the reasons for the security recommendations. This part includes: Securing a Web Server, Securing an Application Server, Securing a Database Server, Securing ASP.NET Applications and Web Services, and Hosting Multiple ASP.NET Applications.
- Assessing Security: This section should show a developer how to evaluate code, designs, and deployed applications for potential vulnerabilities. This section includes: Code Review, and Deployment Review. Microsoft Security Checklists: Contains task-based checklists which are printable quick-reference sheets to help turn information into action. This section includes the following checklists: Architecture and Design Review, Security Review for Managed Code, Securing ASP.NET, Securing Enterprise Services, Securing Web Services, Securing Data Access, Securing a Web Server, and Securing a Database Server.
- Microsoft Security Checklists
- Data Piracy and Data Privacy Standards

Lab Exercises: Labs are needed that demonstrate the key concepts presented above.

Reporting/Logging Technologies: SQL Server

SQL Server Reporting Services are presented in this course. The learner will be able to create reports with calculations, grouping, sorting, matrices, and charts. Managing and programming Reporting Services at the advanced level is also included. The additional subjects of security and creating ad-hoc reports using Report Builder are covered.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Create reports with calculations, grouping, sorting, matrices, and charts
- Manage and program Reporting Services at the advanced level
- Create an ad-hoc report using Report Builder
- Maintain a secure Reporting Services environment

Course Outline

Introduction SQL Server Reporting Services

- Overview of SQL Server Reporting Services Features
- Reporting Services Architecture
- Reporting Services Terminology
- Reporting Services Editions
- Reporting Services Installation

Creating Reports

- Breakdown of Reporting Services
- Shared Data Sources for Reporting Services
- Creating Reports from Requirements

Calculations and Formatting

- Creating Expressions
- Using the Global Collections
- Formatting Items
- Conditional Formatting

Grouping and Sorting

- Creating Groups
- Calculating Totals and Percentages
- Interactive Sorting
- Creating Drill-Down Reports

Report Parameters

- Creating Report Parameters
- Creating Drop-Down Parameters
- Multi-Valued Parameters
- Debugging Parameter Issues

Creating Matrix Reports and Charts

- Creating a Basic Matrix Report
- Matrix Subtotals
- Creating a Basic Chart
- Exploring Charting Possibilities

Managing Reporting Services

- Deploying Reports and Data Sources
- Exporting Reports and Printing
- Using the Web-Based Report Manager
- Using Management Studio Reporting Services Manager

Reporting Services Security

- Reporting Services Security
- Securing Access to Reports
- Data Source Security for Reporting Services

Programming Reporting Services

- Reporting Services APIs
- Reporting Services via URL Access
- Reporting Services Web Service

Advanced Reporting Services Programming

- More on the Reporting Services Web Service
- Working with Custom Assemblies

Snapshots and Subscriptions

- Caching and Snapshots
- Creating Standard Subscriptions
- Creating Data-Driven Subscriptions
- Managing Subscriptions

Ad-Hoc Reporting with Report Builder

- The Report Builder Architecture
- Creating Report Model Projects
- Running Report Builder

Lab Exercises: Labs are needed that demonstrate the key concepts presented above in a scenario that illustrates simple steps in an SDLC: an interview, business requirements, and functional requirements for lab reports.

Inter-Application Architecture: WCF

The fundamentals of Windows Communication Foundation (WCF) are addressed including addresses and binding, service contracts, instance management, data contracts, configuration management, handling errors, WCF Transports, and WCF best practices.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Utilize the topics covered to build and configure a Windows Communication Foundation (WCF) application
- Implement WCF Best Practices

Course Outline

- Introduction to WCF: WCF Basic Terminology and Key Concepts; Address, Binding and Contract, WCF Services and Clients, IIS Hosting, Windows Activations Services, and WCF Architecture Details.
- Addresses and Bindings: Addresses, Bindings, Message Exchange Patterns, Configuring Bindings, Interoperating with ASMX Web Services, Hosting in a Console Application, Channel Factories, and Multiple Endpoints.
- Data Transports: HTTP, Named Piper, MSMQ, and TCP Transport Protocols, Advantages and Disadvantages of the Protocols.
- Service Contracts: Defining Service Contracts (Interfaces), Defining Operation Contracts, Services with Multiple Contracts, Contract Inheritance, and Operation Overloading.
- Introducing XML, SOAP, and JSON for Data Interchange in Web Services
- Instance Management: Using Per-Call Services, Using Per-Session Services, Using Singleton Services, and Configuring Behaviors.
- Data Contracts: Implementing Data Contracts, Mapping Data Contracts to Objects, Serialization, Arrays and Generic Collections, Enums, and Versioning Data Contracts.
- Configuration Management: Using the Service Configuration Utility to configure the service.
- Advanced Service Contracts: Versioning Service Contracts, Implementing Message Exchange Patterns, One-way Contracts, Duplex Contracts, and Asynchronous Proxies.
- Handling Errors: Faults And Exceptions, Generating And Handling Faults, and Fault Contracts.
- Setup and Configuration of IIS for Windows Server
- Service Oriented Architecture (SOA) and AppFabric for IIS Service Bus
- Best Practices for Performance, Transactions and Security.

Lab Exercises: Labs are needed that demonstrate the key concepts presented above in a scenario that illustrates simple steps in an SDLC: an interview, business requirements, and functional and non-functional requirements.

Workflow Modeling: Windows Workflow Foundation

The fundamentals of Windows Workflow Foundation (WF) are addressed including defining the business process workflow, creating and handling workflows, workflow activities, creating and configuring custom activities, applying rules and conditions, managing transactions and compensations, creating and managing runtime services, communications, and using Framework 3.5.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Learn the prerequisites for Windows Workflow Foundation
- Investigate all the available workflow activities
- Learn how to handle exceptions in workflow applications
- Work with Delay and Listen activities
- Learn how to use RuleSets and Policy activities in workflows
- Provide bi-directional communications between the workflow and the host
- Understand the difference between sequential and state machine workflows
- Learn when and how the workflow runtime persists information about a workflow
- Learn when and how the workflow runtime tracks information about a workflow
- Understand how to group activities into a transaction
- Modify an executing workflow from the host application
- Explore how to maintain state between workflow Web service method calls

Course Outline

- Introduction to Windows Workflow Foundation: Key Terminology, WF Architecture, and Visual Modeling of Business Processes.
- Creating and Hosting Workflows: Create State Machine Workflows, Create Sequential Workflows, Select Workflow Authoring Mode, Host a Workflow and Designer in an Application or a Service, Initialize (Passing Parameters) and Manage the Workflow Life Cycle, and Modify a Workflow at Runtime.
- Major Workflow Activities: CallExternalEventActitivity, CodeActivity. ConditionedActivityGroup, DelayActivity, EventDrivenActivity, EventHandlersActivity, FaultHandlerActivity, HandleExternalEventActivity, IfElseActivity, IfElseBranchActivity, ListenActivity, ParallelActivity, PolicyActivity, SequenceActivity, SetStateActivity, StateActivity, StateFinalizationActivity, StateInitializationActivity, SuspendActivity, TerminateActivity, ThrowActivity, and WhileActivity.
- Creating and Configuring Custom Activities: Create custom activities, Create custom composite activities, Define custom activity layout, Add dependency properties, Communicate with the host from a custom activity, and Customize activity serialization.
- Applying Rules and Conditions: Write rule sets, Define rule chaining, priority, and property dependency, Change rules at runtime, Execute rule sets, Implement conditional branching logic.
- Managing Transactions and Compensations: Create and manage transactional workflows, Create and handle compensations, Enlist batch process in a transaction, Perform exception handling, and Handle cancellations.
- Creating and Managing Runtime Services: Create a runtime service, Hydrate and dehydrate workflows, Persist workflows, Track workflows, and Monitor workflows.
- Communicate with Workflows: Handle Events from the Host, Call Methods on the Host, Local Communications, Consume Services from a Workflow, and Expose a Workflow as a WCF Service, Send and Receive Activities.
- Use Framework 3.5 with Send and Receive Activities: Using WF-WCF combined services self-hosted or hosted in a WCF, Communicating with WCF Send and Receive Activities, Maintaining Context Channels, and WF/WCF Conversations.

Lab Exercises: Labs are needed that demonstrate the key concepts presented above in a scenario that illustrates simple steps in an SDLC: an interview, business requirements, and functional and non-functional requirements.

Advanced Programming: C# and VB.NET

Advanced programming topics are covered including C# and VB.NET language features, Windows Presentation Foundation, Silverlight, Microsoft Expression Blend and XAML. The learner will demonstrate knowledge by completing labs focusing on the technical best practices for application performance.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Utilize the skills presented to develop more advanced applications applying best practices for application performance

Course Outline

- Advanced C# and VB.NET Language Options: Includes indexers, dynamic binding, delegation, events, generics, iterators, partial types, implicitly typed local variables, extension methods, Lambda expressions, expression trees, anonymous types, queries, Entity Framework, LINQ, and query expressions and the conventions used to implement these techniques.
- Windows Presentation Foundation (WPF): Details how WPF works and the various ways to use it within .NET applications. Section should cover programming with WPF in C#, and introduce the various class libraries that make up the framework. This section should also introduce the Extensible Application Markup Language (XAML), and show how to combine it with procedural code to build applications that can be deployed to the desktop or hosted in the browser. The topic should cover the new WPF data binding system to hierarchies of objects (master/detail). Demonstrations should be shown for browser-based and Windows WPF applications.
- Silverlight: Includes the building of Windows Vista and Silverlight 2.0 applications by using the new XML-based XAML. This section should also cover application layouts, controls, styles, resources, events, properties, dynamic content, custom components and the application model including navigation and window management. The developer also should be introduced to the new Microsoft Expression Blend tool, and learn how this differs from the desktop WPF that uses desktop tools but can run on non-Windows platforms.

Lab Exercises: Labs are needed that demonstrate the key concepts presented above in a scenario that illustrates simple steps in an SDLC: an interview, business requirements, and functional and non-functional requirements.

Project Lifecycle Coordination for Advanced Software Developers: TFS

This course includes aspects of VSTS including Team Foundation Server (TFS), continuous integration, deployment, and advanced testing. The learner will use graphical modeling tools to create a system design and validate deployment. Other topics include: Managing testing and bug tracking, setting up a project portal for documentation, and performing maintenance. Upon course completion, the learner will be able to create and manage a software project, design a project, generate a code framework, and build a deployment plan, manage test cases, enforce policies and business practices across the project, and customize VSTS to accommodate existing development practices.

Duration: 2 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Create and manage a software project
- Design a project
- Generate a code framework
- Build a deployment plan
- Manage test cases
- Enforce policies and business practices across a project
- Customize VSTFS to accommodate existing development practices

Course Outline

- Introduction to TFS with an overview of Application Lifecycle Management, Requirements, Tasks, Testing, Bugs, Version Control, Templates, Snippets, Code Review, and other concepts
- Best Practices with Team Foundation Server
- Understand TFS tools and how and where they fit into the project lifecycle
- Configure TFS to use different software ALM methodologies like Agile and SCRUM

- Use graphical modeling tools for system design
- Employ TFS for Test Management and Planning
- Use TFS with Microsoft Test Runner
- Use TFS with source control and code check-in and check-out
- Use TFS to manage testing and work items
- Set up portal for project documentation
- Practice Deployment and Maintenance
- Create and manage a software project with TFS

Lab Exercises: Labs are needed that demonstrate the key concepts presented above in a scenario that illustrates simple steps in an SDLC (ALM): requirements, UML and functional and non-functional requirements.

JAVA COURSES

Integrated Development Environment: Eclipse

This introductory course includes an overview of the Eclipse Integrated Development Environment (IDE) and the Software Development Life Cycle (SDLC). Various application features are presented to provide the learner with the skills necessary to navigate within the application and to create a simple data entry form based upon functional requirements.

Duration: 2 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Identify the primary features of the Eclipse IDE
- Install a new perspective into the Eclipse environment
- Create and run a simple Java program within Eclipse
- Utilize Ant within Eclipse to provide code archiving
- Demonstrate familiarity with graphical interfaces
- Understand the significance of the Java "package"

Course Outline

- Eclipse Overview
- Eclipse Menu Bar and Toolbar Features
- Eclipse Views and Panes
- Eclipse Project Explorer
- Eclipse Perspectives
- Eclipse Properties Settings
- Eclipse with Mylyn
- Eclipse Help
- Eclipse Code Editor
- Eclipse Workspace
- Introduction to SDLC
- Eclipse with WindowBuilder
- Creating an Application Window and Widgets
- Handling Window Events
- Eclipse Integration with Ant
- Eclipse Workspace
- Eclipse Project Management
- Window Controls
- Control Events
- Java Class Libraries
- Java Package/Import

Lab Exercises: The attendees will create a simple form based application using Java and Eclipse.

Object Oriented Programming and Design: Java

The main topics of this course include the syntax of properties, variables, methods, modifiers, classes, the principles of object orientation and also its design considerations. The fundamentals from the previous course are utilized to create a more complex solution. Exception handling is demonstrated in a practical way.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Identify the various principles of the object oriented paradigm
- Conceptualize hierarchical relationship of inherited objects
- Recognize various design options in the object oriented paradigm
- Utilize the Java keywords related to object oriented programming in a real programming example
- Understand memory utilization as it relates to object oriented structure
- Define in detail the characteristics of a class
- Utilize the Java keywords and structures related to error handling
- Understand how to recognize and utilize objects in the Java class library
- Understand and use the concepts of reflection and serialization

Course Outline

- Java class / new
- Java Compilation (javac)
- Java package/import
- Java Execution (java) and Classpath
- Java Packages and jar files
- Java Byte Code
- Stack and Heap/Static vs. Virtual Memory Utilization
- Java Class Syntax
- Member Variables
- Properties
- Methods
- Access Modifiers
- Passing Parameters
- Passing Objects as Parameters
- Type Classes
- Object Instantiation and Garbage Collection
- Design Considerations

Lab Exercise: The attendee will create and compile a Java program utilizing the topics presented.

- Generalization & Specialization
- Inheritance
- Containment & Data Hiding
- Polymorphism
- Overloading
- Overriding
- Private Implementation
- Interfaces vs. Abstract Base Classes
- Java interface Keyword

Lab Exercise: The attendee will create and compile a Java program utilizing the topics presented.

- Throwable and Exception Objects
- Java Exception syntax (checked and unchecked exceptions)
- Unhandled Exceptions
- The StackTrace Object
- The try/catch/finally Construct
- Standard Error Types
- Custom Error Types
- Java Reflection Library
- Dynamic Class Loading
- I/O Streams

- Readers and Writers
- Java Serialization Library
- Network Streams
- Recursion

Lab Exercise: The attendees will create an object-oriented program that recursively traverses a tree structure and performs different operations, depending on the contents of an over-riding class.

Data Persistence, Database Connectivity, Entity Modeling: Oracle

This course serves as a formal introduction to relational databases, database construction, and database queries. It includes an introduction to Oracle relational databases, an introduction to XML, an introduction to Java Persistence Architecture and an introduction to the Connection object. Emphasis is placed on successfully creating database queries using SQLDeveloper and then mimicking these queries within a Java program.

Duration: 7 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Design a database in third normal form
- Create a database query using SQLDeveloper
- Perform a database query within Java
- Access data from a database and populate web and Windows forms

Course Outline

- Introduction to Relational Databases, tables, keys, etc.
- Install/Setup Oracle and SQLDeveloper
- Creating an Oracle DB
- Writing Database Queries in SQLDeveloper
- Manipulating Data with Queries
- Select/Insert/Update/Delete with parameters
- Performing a query within Java
- Introduction to XML
- Understanding Transactions
- Creating ERD with SQLDeveloper

Lab Exercise: The attendee will create a simple Oracle database, query the database with SQLDeveloper, and create an entity relationship diagram with SQLDeveloper.

- The Java Connection object
- JDBC (ojdbc.jar)
- Executing Commands and Stored Procedures
- Creating and Passing Parameters through a Query
- Java ResultSet Object
- Java RowSet Object
- Oracle Exception Handling
- Select/Insert/Update Stored Procedures with JDBC
- Handling Exceptions Specific to Data Access
- Java and JDBC Data Types

Lab Exercise: The attendee will execute a stored procedure, pass parameters through a query, and practice handling exceptions.

- Java Persistence Architecture (JPA)
- Using JPA to interact with the Domain Model
- Creating and Mapping a Conceptual Model
- Interacting with Entity Data using JPA
- JPA Data Providers
- JPA Query Language and Criteria API
- Development and Deployment Considerations
- Integration of JPA into an Application
- Integrating data access in a client based program
- Integrating data access into a web based program

Lab Exercise: The attendees will create a Java program utilizing an Oracle database as a data source. The attendees will work as a team to create all of the components needed to satisfy a stated requirement.

Thin Client Development: JSP and Java Server Faces

This course includes introductions to web programming, security, designing the user interface, web forms programming, web services, JavaScript, JSP, Java Server Faces (JSF), JSP AJAX and JBOSS. Using the information obtained in this course, the learner will create a functional, effective and attractive JSP form application with database interaction, data validation and security protocols in place.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Create a functional, effective and attractive web form application with specified database interaction, data validation and security protocols in place

Course Outline

- Servlet Web Programming Example
- Java Server Faces and Facelets Example
- Creating Web Form Applications from Requirements
- Working with Web Objects, Properties and Events
- Federal and State User Interface Standards
- Handling Data within the Request/Response mode
- Catching and Correcting Errors
- Maintaining Security
- Modularization in Web Development
- Creating Custom Web Form Functionality
- Optimizing Web Applications with Caching
- Formatting Web Application Output
- Providing Help
- Internationalization and Localization of Web Applications
- Working with Browser Windows and Events
- HTML Controls, Properties and Events
- Server Controls, Properties and Events
- User Controls, Properties and Events
- Validating User Input at UI and Business Layers
- Formatting with CSS
- Event Handling in Web Forms and Components
- Error Handling of Exceptions
- Handling State: Sessions, Application, and Data Storage
- Unit Testing and Debugging Web Applications
- Introduction to Data Access with Java Server Faces

- Java Servlets as Container Managed Objects
- Web Configuration and Resource files
- Introduction to Web Services
- Creating a Simple Web Service
- Introduction to JavaScript and JSP AJAX
- Web Application Deployment
- Introduction to the Web Server (JBoss)

Lab Exercise: The attendees will create a web application utilizing Java Server Faces using a requirement specification and the web server environment provided.

Thick Client Development: Eclipse

This course includes introductions to GUI programming with Eclipse, security and designing the user interface. Using the information obtained in this course, the learner will create a functional, effective and attractive form application with database interaction, data validation and security protocols in place.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Create a full scale Java application using Eclipse, including debugging and validation
- Interpret requirements to successfully implement user needs
- Create interfaces that satisfy federal and state guidelines
- Understanding error correction at the user interface level
- Deploy a Java application to a non-development environment

Course Outline

- Introduction to GUI Programming with Eclipse
- Creating GUI Forms Applications from Requirements
- Working with GUI Forms Events and Properties
- Federal and State User Interface Standards
- Accessing Data with JDBC in Swing Applications
- Catching and Correcting Errors
- Maintaining Security
- Modularization in Application Interface Development
- Creating Custom Window and Widget Functionality
- Applications with Multiple Windows and Dialogs
- Java Text Formatting Utilities
- Providing Help
- Federal and State Standards and GUI Applications
- Working with Windows and Events
- Controls, Properties, and Events
- Validating User Input at UI and business layers
- Formatting with Swing
- Event Handling in Swing Windows and Controls
- Error Handling in Swing Applications
- Unit Testing and Debugging Swing Applications
- Introduction to Data Access using JPA in Swing Applications
- Swing Programming and the Library Class Hierarchy
- Swing Application Configuration and Resource files
- Introduction to Smart Client Applications
- Introduction to Office Automation
- Java Application Deployment and Distribution

Lab Exercise: The attendees will follow stated requirements to create a full-featured Java application. A set of components will be provided so that the experience includes integration as well as new code. Detailed validation activity will be required.

Application Security: Java EE 6

This is a comprehensive course on system security and data privacy issues. The learner will be able to assess existing infrastructure and applications, identify threats and vulnerabilities and be able to eradicate them. Methods for designing and developing new secure web applications will be presented in detail. Data privacy standards and policies will be discussed. Emphasis will be placed on how to develop applications that are compliant with the existing standards and policies.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Assess existing infrastructure and applications
- Identify threats and vulnerabilities
- Eradicate security threats and vulnerabilities
- Employ methods for securing web applications
- Protect data from piracy and adhere to data privacy standards

Course Outline

- Assessing Security
- Basic Threat Terminology
- Threats and Counter Measures
- Designing Secure Web Applications
- Building Secure Web Applications
- Securing a Host and Application
- Java EE 6 Security Checklists
- Data Piracy and Data Privacy Standards

Lab Exercise: Given a scenario of a system with vulnerabilities, the attendees will work as a group to identify the threats and vulnerabilities. Given a scenario of a web application with vulnerabilities, the attendees will work as a group to identify the threats and vulnerabilities.

Reporting/Logging Technologies: Jasper

An overview of the Jasper Reports Library is presented in this course. The learner will be exposed to all the major features of this reporting tool and will gain practice in producing reports of various formats, from a variety of data sources.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Create a report, populate it with data and represent it in a variety of formats including print/display, PDF, HTML, RTF, XML, TXT, etc.
- Create reports containing mathematical expressions and statistical representations
- Understand the structure of the Jasper Library API
- Discriminate between good and bad report designs
- Create reports containing graphical data
- Configure Jasper Reports to create custom and “ad hoc” reports

Course Outline

- Introduction to the Jasper Reporting Library
- API Overview
- Report Designs
- Compiling Reports
- Populating Reports with Data
- View, Print and Export Reports
- Parameters
- Data Source
- Fields
- Expressions
- Variables
- Report Sections
- Frames
- Fonts and Unicode Support
- Styles
- Scriptlets
- Subreports
- Internationalization
- Datasets
- Charts
- Groups
- Crosstabs

Lab Exercise: The attendees with model a census data collection application and provide output in a variety of formats as defined by the exercise.

Data Exchange and Configuration Support: Java

The fundamentals of Java Naming and Directory Interface (JNDI) are addressed as well as an introduction to XML, SOAP, and JSON for Web Services. Other topics include: configuration management, setup and configuration of JBoss for Windows Server, and service-oriented architecture (SOA).

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Utilize JNDI to access multiple naming and directory services.
- Understand serviced-oriented architecture (SOA)
- Encode an HTTP header and an XML file using SOAP
- Represent simple data structures and objects in a web browser using JSON
- Setup and configure JBoss for Windows Server

Course Outline

- Overview of Java EE 6 and Java Core I & II
- JNDI
- Data Transports (HTTP, MSMQ, etc.)
- Service Contracts
- Introduction to XML, SOAP, JSON for Web Services
- XML Data Exchange Protocols
- Configuration Management
- Advanced Service Contracts
- Handling System Errors

- Setup and Configuration of JBoss for Distributed Data Environments
- Service Oriented Architecture
- Best Practices for Performance, Transactions and Security

Lab Exercise: The attendee will practice using JNDI, SOAP, and JSON for Web Services in hands-on lab exercises. In a group setting, the attendees will assess good and poor examples for best practices in performance, transactions and security.

Workflow Modeling: IMIXS

The fundamentals of the IMIXS Workflow Engine and Workflow Modeling Tool will be addressed including principles of business process workflow, how to create an appropriate workflow model, how to represent the model and then to implement it using the IMIXS Workflow Engine.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Analyze a business process and conceptualize a workflow pattern that will accurately model that process.
- Represent a workflow design in the IMIXS Workflow Modeling Tool.
- Install and configure the IMIXS Workflow Engine in a JBoss Application Server environment.
- Implement a workflow model using the IMIXS Workflow Engine including all required behaving and role definitions, and communication with the host application.
- Utilize additional IMIXS tools and features designed for the efficient development and management of a Workflow System.

Course Outline

- Introduction to business process workflow modeling principles and key terminology
- Designing workflow solutions and defining patterns of business process activity
- Implementing workflow variants: State Machines, Sequential Workflows, Composite Workflows, etc.
- Managing a workflow implementation: managing the workflow lifecycle, runtime modification, initialization, etc.
- Creating workflow rules
- Communication between the workflow engine and the host application

Lab Exercise: The attendees will design and represent a workflow solution for given user requirements. The attendees will implement a pre-defined workflow within a web application using IMIXS.

Advanced Programming: Java

Advanced Java programming topics are covered. The learner will demonstrate knowledge by completing labs focusing on the technical best practices for application performance.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Utilize the topics covered to create complex solutions adhering to best practices for application performance

Course Outline

- Advanced Java Language Topics
- Annotations
- Java Native Interface
- Collections

- Java I/O
- Dependency Injection
- Aspect Oriented Programming
- Enterprise Java Beans
- Concurrency
- Generics
- Custom Generics
- Additional Topics to be Specified

Lab Exercise: The attendee will create a complex solution using the topics covered, adhering to best practices for application performance.

Project Lifecycle Coordination for Advanced Software Developers: Maven

This course includes fundamental aspects of the Lifecycle Management with Maven.

Duration: 3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Have a practical understanding and experience of Team Foundation Server (TFS)
- Have a practical understanding and experience of Maven
- Understand the issues around dependency management in Java development, and be able to implement declarative dependency management in a corporate environment using both Maven and Ant
- Improve developer productivity though the use of standards and conventions
- Improve team communication with the automatically-generated Maven web site
- Design more modular and more reusable software

Course Outline

- Lifecycle Management with Team Foundation Server (TFS)
- Best Practices for Project Lifecycle Management
- Lifecycle Management with Maven
- SCRUM Influences on the Lifecycle Management Process
- UML and Other System Design Documentation
- JUnit and Other Test Management Tools
- Use TFS as a Test Management Console
- Use TFS with Ant and Maven
- Alternative Test Management Consoles
- Set up a Portal for Project Documentation
- Deployment and Maintenance
- Managing a Software Project with Support Tools

Lab Exercise: The attendees will set up a project documentation portal, practice deployment and maintenance, and create and manage a software project with support tools.

Presentation Skills for Java Programmers

Visual Presentation will have increasing importance as Tennessee's IT operations move into higher levels of sophistication. Modern design concepts are more visually oriented (class hierarchies, ERDs, dynamic data, etc.) and the tools for creating better presentation artifacts are less expensive and easier to use. This course will introduce a variety of presentation formats, tools and techniques that are available specifically to the Java oriented developer. In addition to the usual communication needs, the integration of these tools with other development processes (such as test result reporting) will likely become more prevalent as these skills increase.

Duration: 2-3 Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Select the best presentation mode for the situation
- Select the best presentation tool for the situation
- Utilize the selected tool to produce an effective visual presentation

Course Outline

- The Importance of Visual Presentation
- Overview of Presentation Options
- Microsoft Office
- PostScript Document (PDF)
- Visio for UML
- Java Graphics
- Java FX
- Window and Web Application as Prototyping Agents
- Mockingbird
- More Swing/XWT/WindowBuilder
- Gnuplot

Lab Exercises: Using a provided framework, implement an automated task to create a chart in Excel using comma delimited data and also paste that chart into a Word Document. Using a provided specification, use Mockingbird to prototype a web form. Then, use a provided mark-up of that same form and implement the changes in the prototype.

ORGANIZATIONAL / PROCESS COURSE DESCRIPTIONS

Organizational and Process Oriented courses are interdisciplinary courses that will be provided for all IT professionals in all State Agencies. The goal is to develop a common set of expectations and a framework for working together on IT projects regardless of which department and what project.

State of TN IT Professional Orientation Modules

These three (3) State of TN IT Professional Orientation Modules are provided in the Associate level training for all IT professionals and they would also be terrific courses to assist with training business professionals working within State Agencies, and Executives within the State who participate in planning for IT projects. Together these modules will be used to support the cultural shift within the State of TN IT environment to the desired culture of the NextGen IT Transformation. All three modules would become computer-based curriculum once the transformation is well underway. They will be taught in a classroom setting until there is evidence that the transformation has taken hold. The classroom setting will allow for the type of Q&A dialog that could be useful in supporting this transformation.

The specific course content will need to be closely coordinated with the State NextGen IT Transformation initiative. This is not an off-the-shelf series of modules. The learning objectives and course outlines below serve as a structure for the courses. Instructors must work closely with the State to communicate the specific messages of the initiative.

Duration: Each Module will be approximately 1 hour

Audience: All IT personnel, Agency leadership, Executives participating in IT project planning, may be relevant for vendors working in support of IT projects, products, or services.

Module #1: Orientation to TN IT Governance and Customer Service Mission

Learning Objectives

Upon completion of this course, attendees will be able to:

- Describe how IT operations and service delivery is organized within the State of TN
- Describe the types of products and services State Agencies can receive from State IT including OIR and BSD
- Describe the importance and practice of the customer-service mission
- Describe their personal responsibilities for working in the State of TN NextGen IT environment.

Course Outline

- Overview of the State of TN IT Governance
  - Offices, Departments providing IT services and products
  - Relationship between Agencies and State IT Departments and Offices
  - Role of Vendors in supporting IT products and services
  - Role of IT in working with Vendors
  - Introduction to OIR and BSD, and the types of services and products Agencies are able to receive from these State IT offices.
- The Customer-Service Mission - serving the people of the great State of TN
  - Who are State IT customers?
  - The importance of personal responsibility, integrity, and discipline in all IT professionals.
- The NextGen IT Environment
  - Expectations for team work, quality, collaboration, respect for others, and professional behavior
  - Day-to-day, work-specific impact of the customer-service mission on project teams, and IT professionals.
  - IT professional development plans including the new learning program

Module #2: Orientation to TN Business Solutions Methodology (TBSM)

The purpose of this module is to establish a common understanding of the life of a project within the State of TN, and to increase understanding of how the participants in the class fit into the project team, no matter if they are part of the technical team, a customer, an end user, a business manager, or a vendor.

Learning Objectives:

Upon completion of this course, attendees will be able to:

- Describe at a high level how projects are planned, designed, executed, and operationalized within the State.
- Describe the primary types of projects the State of TN is likely to plan and execute
- Recognize where the work they typically do fits into the project lifecycle
- Know where to get more information on the project lifecycle

Course Outline:

- High level introduction to the TBSM
- Why we have the TBSM
- How to assess support from BSD to further understand the methodology and use templates
- Different Types of State projects and how to approach these projects using the TBSM

Module #3: Orientation to TN Standard Software Development Process

Software Development Processes provide a framework for all software development projects regardless of size, complexity, and risk. The focus of this course is to establish an expectation for all software development projects to utilize a basic set of processes to ensure quality deliverables.

Learning Objectives:

Upon completion of this course, attendees will be able to:

- Describe the expectation that all projects use standard processes and best practices
- Describe in general terms how the processes are applicable no matter what SD methodology is implemented on a project.
- Identify the SD processes and give a high level description of the purpose of the process
- Know where to get more information on the SD process

Course Outline:

- Fundamentals of State IT Standard Software Development Processes
- Purpose behind State establishing minimal processes that are required for all SD projects
    - Benefits of using the State of TN Standard Software Development Processes
    - Examples of the kinds of trouble projects can experience when standard process is not utilized to structure the work.
- Difference between a Project lifecycle, Software Development Lifecycle, and the State's Standard Software Development Processes, and how they synch-up, (example where the SD process would fit in a Project lifecycle).
- High level description of each process and where it is used in the Software Development Lifecycle
- How to get additional training when needed

Team Roles and Responsibilities

This course is designed to build on the orientation modules. Building on the Standard TN Project Lifecycle and TN Software Development Lifecycle, the course will further define the roles and responsibilities of team members working on a project. The course will highlight the different types of projects that the State might undertake, and how the project teams vary in size and complexity. There will also be discussion of the importance of adapting in smaller

projects and even more carefully defining the roles and responsibilities of the team members because smaller projects often do not have the level of planning and oversight that larger projects receive.

Since this is a Junior level course, it will not go into depth about how to pull a team together or how to lead a team. The objective is for the participants to increase their understanding of the workings of a team and to reinforce the importance of stepping-up, taking personal responsibility for working as a part of the team by responding to the leadership within the team and giving their personal best to the team.

Duration: 3 hours

Learning Objectives

Upon completion of this course, attendees will be able to:

- Recognize varying size and complexity of team structures
- List the characteristics of effective team members
- Identify the roles and responsibilities found within teams
- Clarify roles and responsibilities on the team
- Respond to team leadership effectively

Course Outline

- Planning for success
  - What is a team?
  - Task oriented vs. team oriented behaviors
  - Team positioning
- Defining team roles
  - Productive team member roles
  - Destructive team member roles
  - Team accountability
  - Individual accountability
  - Interaction with team leadership
  - Monitoring team progress

Lab exercise: Participants will be given the opportunity to discuss the differing roles and responsibilities of the team members and share insights from previous team memberships both good and bad. Instructors will lead the conversation and be certain to include the topics covered in the course outline.

Time Management

For Junior level Software Developers who are becoming more knowledgeable about what they can do with the technology but still having many new experiences, the ability to monitor their own use of time becomes more important. At this stage, they will be producing code for projects that have schedules and due dates for deliverables. They need to begin to know how to manage their time, focus their activities, and accurately evaluate the time needed for various tasks. This course will be a lab where participants will be given an opportunity to practice using techniques for organizing their work to increase efficiency and gain awareness through timed activities about how long tasks take for them individually and for the group. It would be nice to make this interactive and to include some opportunities for participants to share their own tips for improved accuracy and timeliness. It will be critical for the Instructor to set ground rules. The goal is not to be the fastest developer – the goal is for each person to know their own speed, and to know the optimum ways they personally produce the best quality within the timeframes provided.

Duration: One Day

Learning Objectives

Upon completion of this course, attendees will be able to:

- Set S.M.A.R.T. goals
- Prioritize goals effectively

- Understand the needs of different personality styles and how to work with them
- Handle high pressure, crisis situations
- Prioritize time and tasks effectively
- Achieve better results through effective planning
- Overcome procrastination
- Estimate time and activities required for reaching goals
- Handle paperwork effectively
- Manage resources more efficiently
- Organize workspace
- Use time management tools more effectively
- Become effective at delegating for maximum productivity

Course Outline

- Working with Goals
  - S.M.A.R.T. (Specific, Measurable, Attainable, Relevant, Time-Bound) Goals
  - Prioritizing Goals
- Working with Others
  - Personality Types A, B, C, and D
  - Recognizing and Working With Different Personality Types
  - Team Dynamics that Affect Timely Deliverables
  - Handling Crisis Situations and Project Delays
- Prioritizing Time & Tasks
  - The 80/20 Rule
  - The Urgent/Important Matrix
  - Assertiveness
- Planning Wisely
  - Creating a Productivity Journal
  - Glass Jar: Rocks, Pebbles, Sand and Water
  - Estimating Time and Activities
  - Processing Required Paperwork
  - Managing Available Resources
- Organizing Workspace
  - De-clutter
  - Managing Workflow
  - Dealing with E-mail
  - Using Calendars
- Delegating Made Easy
  - When to Delegate
  - How to Delegate
  - Managing Responsibility when Delegated

Lab Exercise: Participants will be given the opportunity to practice using techniques presented for organizing their work to increase efficiency and gain awareness through timed activities of how long tasks take for them individually and for the group.

Foundations of TN Standard Software Development Processes

This series of modules will provide a deeper dive into the processes. These modules will be 1-2 hours in length and can be web-based. They will result in participants being able to carry out tasks within the process areas with supervision from the Lead Developer or Architect. The content should be relevant regardless of the software development lifecycle meaning that there are tools, techniques, practices that are relevant and tailor-able regardless if the development model is waterfall or agile or rapid prototyping. The difference is these approaches can be explained in the modules and even the benefits of using the different approaches and tools depending on the type of project.

Duration: Modules will be 1-2 hours in length – Web-based

Learning Objectives

Upon completion of this course, attendees will be able to:

- Carry out tasks within the process areas with supervision from the Lead Developer or Architect

Course Outline

- Analysis
- Defining Requirements
- Design
- Estimation
- Establishing and Migrating between Development, QA, and Production Environments
- Configuration Management
- Change Management.
- Testing and Verification
- Deploying to Production
- Production Operations
- Enhancements and Maintenance

It will be important to point participants toward tools and resources that can be used just-in-time because Junior developers will not engage in these activities on a daily basis and they will lose what they learn unless they have access to resources when needed.

Project Management Basics

This course is designed to give the Lead Developer some management skills that will help structure and improve their ability to manage their tasks. Basic Project Management phases and skills will be addressed. The course will integrate process, communication, leadership, problem-solving and decision-making skills to ensure there is an awareness of the inter-disciplinary aspects of leading a project. The course will also refer to TN IT Governance and the Tennessee Business Solutions Methodology and help Leads understand how their work integrates with the bigger picture of the full project scope.

Duration: One Day

Learning Objectives

Upon completion of this course, attendees will be able to:

- Explain the importance of clearly defining a project, identifying sponsors, and generating buy-in.
- Identify barriers to running successful projects.
- Apply techniques for dealing with scope creep and changing priorities.
- Demonstrate how to ask effective questions to identify what's really important.
- Effectively apply techniques for running productive project meetings.
    - Develop an action plan to address current project shortcomings.

Course Outline

- Defining a Project
    - Putting Together the Project Team
    - Asking the Right Questions
    - Identifying Stakeholders
    - Identifying Importance and Priority
- Developing a Schedule
    - Identifying Barriers
    - Milestones
    - Key Tasks and Dependencies
    - Determining Resource Requirements

    - Tracking with Gantt, PERT
- Managing the Expected and the Unexpected
    - Identify risk
    - Dealing with Expanding Requirements and Changing Priorities
    - Requirements Documentation
    - Building an Issues List
    - Writing Status Reports
    - Creating an Action Plan to Address Issues
- Coordinating a Project Team
    - Techniques for running Project Meetings
    - Assigning Responsibility
    - Following up with Team Members
- Maintaining Relationships
    - Lab exercise in solving issues related to communication problems
- Project Completion
    - Project Debriefing
    - Documenting issues for future projects

Lab Exercise: Participants shall take a small "project" and discuss forming the project team, developing the schedule, identifying risks, dependencies and dominant communication channels.

COMMUNICATION COURSE DESCRIPTIONS

Note: The instructors will be very skilled at establishing a safe, collaborative environment themselves so that the role-play and exercises are non-threatening and successful. In all courses such as this one the instructors must recognize that they are teaching technical persons these 'soft' skills that do not always come as naturally as technical skills. Participants are likely to be reluctant and maybe intimidated to participate in role-play and interactive activities.

Introduction to Technical Communication

This course will provide the Associate Developer with the most basic communication skills – writing and speaking, active listening, verifying what has been communicated. The context will be technical but the objective will be to shore up core communication skills and awareness of the importance of really communicating effectively in a technical environment. Course designers should take care to make the course applicable to the adult learner, keeping the learning interactive and engaging so that all learners will recognize some value and not leave with the impression that the course was too juvenile.

Duration: One to Two Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Define why listening is important
- Demonstrate the use of open-ended and closed questions
- Describe a model of feedback, communication, and listening
- Explain the importance of body language
- Effectively introduce themselves and others
- Rephrase blunt language for better communication
- List techniques for dealing with difficult people
- Develop an action plan to improve communication skills

Course Outline

- Communication as a Tool for Technology Professionals
  - Why Communication Skills are Critical with Other Technical Professionals
  - Why Communication Skills are Critical with Non-Technical Professionals
- Active Listening
  - How to Listen for Important Content
  - How to Respond to the Person Talking
  - How to Ask Questions Effectively
  - Verifying What Has Been Communicated Verbally
- Presentation Skills
  - Using Appropriate Affirming Body Language
  - Making an Introduction
- Rephrasing for Better Relationships
  - Phrases to Avoid
  - How Can the Point be Better Received
- Voicemail, Email, Memos, and More
  - How to use Office Communication Tools for Maximum Effectiveness
  - Verifying What Has Been Communicated in Written Form
- Introduction to Communicating Technical Information to Non-Technical People
  - How to Communicate Complex Information to Laymen
- Dealing with Challenges
  - Case-study evaluations of ways to deal effectively with difficult personalities and tough situations. From negative people, "backstabbers", whiners, to minimal contributors, participants will discuss better ways in which to communicate and work with those whose actions make the process hard.

Lab Exercises: This course has several labs including: practices in listening for comprehension, practices in writing emails and memos, practices in wording technical jargon for receipt by laymen, and those mentioned in the outline concerning dealing with challenges. The participant will also create an action plan to improve their personal communication skills.

Technical Documentation Skills

For the Junior Developer it is important to further refine the skills of technical documentation. This course will combine technical and communication concepts to increase participant ability to write useful comments in code, useful technical documentation regarding the software, and useful user documentation for printed and electronic versions of help. Attention will be given to how to speak 'non-technically', how to break a process into a logical flow, and the importance of using consistent terminology regarding common instructions such as clicking, selecting, buttons, links, pages, saving, deleting, etc. for the full project system – ensuring that the written communication will be successfully understood by someone who has not worked on the system.

Duration: One or Two Days

Learning Objectives

Upon completion of this course, attendees will be able to:

- Identify different types of technical documents
- Adapt writing style to meet the intended audience's knowledge level
- Organize information in a logical way
- Structure a technical document in a logical way
- Adhere to organizational guidelines/requirements for technical documentation format and content
- Follow existing template and formats
- Insert graphics and tables into a technical document in appropriate places
- Check and edit technical documents written by the attendee or by others
- Prepare a technical document for presentation or publication
- Use multimedia to disseminate the technical document

Course Outline

- Different types of Technical Documents
  - Requirements Documents
  - Design Documents
  - Software Source Code Comments
  - Software Revision History/Logs
  - Log Messages
  - Software Error Messages/Help Documentation
  - Software End User Documentation
  - Project Documentation
  - Tutorial Documents
- Why Are You Writing, and Who Is Your Audience?
  - Defining the purpose of the document
  - Keeping the reader in mind
- Planning Your Document
  - Organizing the information into a logical and manageable structure
  - Different formats and conventions for technical documentation
  - Setting up the plan and keeping track of progress
- Writing for Clarity and Impact
  - Maintaining consistency and flow of meaning
  - Maintaining organizational requirements for syntax and content
  - Using plain English when possible

- Use of Words, Graphics, and Tables
  - Dealing with jargon, technical words, acronyms and abbreviations
  - Consistency in Wording for Same Task or Idea
  - Incorporating diagrams where appropriate
  - Checking and Editing
  - Checking for Accuracy and Flow
- Presentation and Publication (for Published Documents only)
  - Preparing the document for publication
  - Techniques for design, layout and typography
  - Multimedia and Electronic Information Management

Lab Exercises: The exercises will include practices in writing snippets of each of the different types of technical documents efficiently and effectively utilizing all of the topics presented.

Developing a Customer Service State-of-Mind

The focus of this course is to reinforce and further develop awareness and ability to work with the goal of service in mind. Many of the State's IT personnel do not have face-to-face interactions with the end users or 'the people' so the idea of giving them training regarding customer service may seem less than useful. The reason for this course is that customer service is a mindset – an approach to daily work as well as skills and abilities to engage and serve another person. This course will focus on both aspects.
Objectives:

- Reinforce the mindset of customer service as the driving motivator behind State IT activities – identify barriers to these critical values for the State IT environment such as pride in work, strong work ethic, effective solutions, cost effective results.
- Learn ways to actualize and internalize these values personally and professionally.

The format for the course should be interactive and dynamic with opportunities for role play, perhaps watching scenarios of worker interactions and discussion of how the values transfer to the work environment and the work produced.

Duration: ½ Day

Learning Objectives

Upon completion of this course, attendees will be able to:

- Recognize that customer service is the driving motivator behind State IT activities
- Identify the customer
- Identify barriers to the critical values such as pride in work, strong work ethic, effective solutions, cost effective results
- Actualize and internalize these values personally and professionally
- Utilize these attributes to fully address customer needs and to maximize customer service across all IT job classifications

Course Outline

- Who We Are and What We Do
  - Who are customers (internal/external)?
  - What is Customer Service?
  - Who are Customer Service Providers?
- Establishing a Positive Attitude
  - Appearance Counts (even if not in person)
  - Staying Energized
  - Staying Positive

- Be the Best You Can Be
  - Establishing a Strong Work Ethic
  - Developing Self-Assurance and Appreciating a Sense of Accomplishment
  - Maintaining an Awareness of Costs and Desiring to Minimize Where Possible
- Identifying and Addressing Their Needs
  - Understanding the Customer's Problem
  - Staying Outside the Box
  - Meeting Basic Needs
  - Going the Extra Mile

Lab Exercises: The format for the course should be interactive and dynamic with opportunities for role-play, perhaps watching scenarios of worker interactions and discussion of how the values transfer to the work environment and the work produced.

Team Dynamics

The Team Dynamics course is for the Intermediate Developer who is functioning with less supervision and is now being required to take more responsibility for the success of the team. The course should include content related to working with Junior and Associate Developers, and Leads. This is an opportunity for an engaging workshop where roles and responsibilities can be reinforced. Participants should come away with a basic understanding of how to be a positive, contributing member of a team. They should have a basic understanding of team etiquette, boundaries, flexibility, and respect for each other. This is a course that all personnel will participate in at some point and it would be ideal to have it be multi-disciplinary so that role-switching and team tasks can be completed in the classroom.

Duration: ½ Day

Learning Objectives

Upon completion of this course, attendees will be able to:

- Understand how to promote collaboration
- Successfully collaborate with others
- Effectively handle team diversity
- Recognize positional boundaries and when it's appropriate to aspire to a new position
- Maintain flexibility in work schedule and project deliverable deadlines
- Identify areas for self-improvement
- Identify areas for team improvement

Course Outline

- Team Etiquette
  - Considerate Workplace Conduct
  - Respecting Team Members
  - Dealing with Diversity
  - Identifying and Addressing Sources of Conflict
  - Planned Intervention
- Boundaries
  - What are Positional Boundaries?
  - Vying for Position
- Flexibility
  - Adjusting Schedules and Priorities to Fit Changing Needs/Goals
- Self-Assessment
  - What Can Be Done To Make Me More Effective to the Team?
- Team Assessment
  - What Can Be Done To Make the Team More Effective?
  - Optimizing Team Performance

Lab Exercises: The lab for this course will include role-playing with role switching and simple project team tasks.

Improving Understanding between Technical and Non-Technical Project Stakeholders

The Intermediate Developer is becoming highly technically fluent yet the need to speak non-technically is increasing. This course will help the participants recognize the change in their ability to talk effectively to a non-technical person. The best approach to this course would be to have mini-movies of examples of ineffective and effective interactions between technical and non-technical people that the class can discuss, followed by role-playing verbal exchanges – especially using role-swapping exercises. Techniques for interpretation, translating, verifying understanding, preparing for presentations / meetings, and facilitating discussions should be presented and practiced.

Duration: ½ Day

Learning Objectives

Upon completion of this course, attendees will be able to:

- Communicate effectively with non-technical colleagues
- Illustrate methodologies or concepts when appropriate to aid in recipient understanding
- Present information effectively while actively listening to feedback
- Adjust communication as needed to further clarify ambiguous subjects/topics
- Present non-technical colleagues with understandable technical options

Course Outline

- How to use Analogies Effectively
- Limit Your Message to "Need-to-Know" Information
- Display Your Thinking Visually
- Involve the Listeners
- Manage the Intimidation
- Structure Your Message Effectively
- Avoid Projecting Opinion as Fact
- Helping Non-Technical People Make Sound Technical Decisions

Lab exercises: The exercises for this course will provide the attendees with opportunities to create communication documents (emails, memos, presentations) in a manner that non-technical staff can comprehend. Short verbal presentations and/or conversations will also be role-played.

Technical Communications for Leads / Managers

This course will help the Advanced Developer become more proficient in technical documentation, communication with management and other stakeholders, and successful communication with developers and other technical team members who are seeking direction from the Lead developer. Again the focus will be written and verbal communication. It should address the individual's own communication as well as how to guide others in their communications, how to provide feedback, how to edit other's documentation.

Duration: One Day

Learning Objectives

Upon completion of this course, attendees will be able to:

- Determine the content of your message
- Determine the best way to prepare and deliver your message verbally
- Examine the basics of building a well-structured presentation
- Examine the mechanics of delivering a successful presentation
- Manage team communications effectively

Course Outline

- Write to be Understood
  - Determine When to Write
  - Plan Your Message
  - Perfect Your Style and Mechanics
  - Writing in the Workplace
- Speak to be Heard
  - Prepare Your Message
  - Deliver Your Message
- Plan for High Impact Presentations
  - The Importance of Planning
  - Outline Your Presentation
  - Develop Your Script
  - Add Visual Aids
- Captivate Your Audience
  - Make Final Preparations
  - Basics of Delivery
  - The Question-and-Answer Session
  - Assess the Presentation
- Holding Effective Meetings
  - Clarify the Purpose of the Meeting
  - Create the Agenda
  - Lead a Session Effectively
- Managing Team Communications
  - Creating a Communications Plan
  - Determining what Tools will be used for Documentation and Communication
  - Providing Feedback
  - Editing Documents Produced by Others

Lab Exercise: The attendee will plan and present a short presentation given a specific topic.

Advanced Team Dynamics

The focus of this course is to engage Advanced personnel in techniques for working through difficulties and challenges of dysfunctional teams. *This course has the pre-requisite of completing the Team Building course in the Leadership Track prior to participating in the class*. It will be interactive and will address ways to overcome common problems such as poor performance, differing opinions, lack of communication, and lack of leadership.

Duration: One Day

Learning Objectives

Upon completion of this course, attendees will be able to:

- Define and understand the sources of conflict
- Resolve conflicts using different strategies
- Identify their own personal conflict resolution style
- Understand the different bases of power and how to change them
- Apply influence and explore their relationships with others
- Provide conflict management training for others

Course Outline

- Definitions of Conflict
  - Misconceptions about Conflict
  - Sources of Conflict

    - Positive and Negative Factors of Conflict
    - Business Management and Conflict Resolution
- Conflict Mode Instrument
    - Scoring and Interpretations
    - Ways of Coping with Conflict
    - Assumptions and Outcome of Conflict
- Influencing Others in a Problem-Solving Context
    - Working Effectively with Team Members
    - Managing Your Emotions, Information and Problems
    - Tips For Effective Day to Day Conflicts
    - Resolving Conflict before It Gets Out of Hand
    - Managing Conflicts with Superiors and Subordinates
- Importance of Team Work
    - Dealing with Dysfunctional Team Roles
    - Managing Conflict in Teams
- Influence Inventory (Power Bases)
    - Definitions of Influence and the Bases of Power
    - Changing the Bases of Power
    - Leadership Training for Influence and Power
    - Training Development for Influence

Lab Exercises: The lab for this course will be interactive role-playing through situations presented in the exercises. Some of the situations included will be: slacking team member, conflict resolution, lack of communication, and lack of leadership.

PRACTICAL REASONING COURSE DESCRIPTIONS

Courses in this area of competency will expand upon the professional's ability to actively and skillfully conceptualize, apply, analyze, synthesize, and/or evaluate information gathered and to use these skills to create more effective, efficient, and appropriate deliverables; solve problems, and make decisions as necessary within their job classification.

Resources to Improve Practical Reasoning

This set of training modules will give State IT personnel access to ideas, tools, and methods to improve their ability to think critically, understand and solve problems more effectively, and make informed decisions. Many training vendors have extensive curriculum related to these topics so these modules are likely to come from a vendors existing course catalog. The lists below provide an example of the kinds of modules the State would like to have available for their personnel. The actual offering from a training vendor may be more extensive. These can be instructor-led, computer based, or lab courses.
Duration: Each module in a series will be a stand along module and should be no more than 1 hour.
Audience: These modules will be applicable to any IT personnel including those who are technically focused and those who are business focused. These modules will be available to anyone as electives that they can select on their own or as directed by their managers as part of their Individual Development Plan.

Critical Thinking Series

Learning Objectives

Upon completion of one or more modules, participants will be able to:

- Recognize how to think more logically,
- Identify ways to apply critical thinking methods to improve their own skills
- Identify ways to use critical thinking skills in their work

Course Outline

Topics for Modules will include but are not limited to:

- Organizing information – learning ways to organize information in a variety of ways
- Exploring ways to identify many possible options or alternatives to how information can be organized
- Methods for evaluating alternatives
- Evaluating data and finding the data most relevant to the task
- Drawing conclusions from information
- Identify what has not been stated when analyzing instructions, or steps in a process
- Developing originality and creative thinking
- Applying critical thinking techniques in the workplace
- Self-awareness – identifying strengths, and weaknesses, improving your own abilities.

Problem Solving Series

Learning Objectives

Upon completion of one or more modules, participants will be able to:

- Have an increased awareness of problem solving methods, tools and approaches
- Identify ways to use problem solving techniques to improve their work

Course Outline

Topics for Modules will include but are not limited to:

- Basic steps to solve a problem
- Finding and choosing tools and techniques to improve problem analysis
- Identifying and overcoming barriers to problem resolution
- Techniques for identifying and evaluating possible solutions
- Planning to implement a solution including ways to identify possible undesired outcomes and plan for contingencies
- Implementing a solution
- Evaluating the solution

Decision Making Series

Learning Objectives

Upon completion of one or more modules, participants will be able to:

- Recognize how to approach decision making more deliberately
- Understand what information may be needed to make a good decision
- Recognize that there are tools to help with different types of decision making
- Identify ways to improve their own decision making in the workplace.

Course Outline

Topics for Modules will include but are not limited to:

- Approaches to decision making
- Finding and choosing tools and techniques to structure and support decision making (for example, Financial decision tools, making decisions with limited or incomplete information)
- Identifying and dealing with your own biases

Using Critical Thinking, Problem Solving and Decision Making Techniques to Improve Deliverables

This course will be an experience-based lab designed to give attendees the opportunity to actively and skillfully conceptualize, apply, analyze, synthesize, and/or evaluate information using critical thinking and problem solving techniques and methods. They will be moderately complex scenarios that are designed for intermediate - senior level developers.

Duration: 1 day

Audience: Developers, DBAs, Business Analysts

Learning Objectives

Upon completion of one or more modules, participants will be able to:

- Demonstrate the ability to use critical thinking techniques and methods to improve their designs and approaches to development tasks
- Identify their strengths and know how they can improve their skills

Course Outline

Problem-Solving and Decision Making

- Defining the Problem
  - Identifying the real problem
  - Finding true causes
  - Formatting the problem
  - Using a problem-solving method
  - What, where, when, extent
  - Distinctions between causes
- Defining the Objective
  - Mapping to the statement of the problem
  - Characteristics
  - Measurable: using an explicit format
  - Clarity
  - Relevant: showing the right content
  - Alignment with organizational objectives

How to Generate Alternative Solutions

- Role of creativity: thinking outside the box
  - Left vs. Right brain thinking
  - Convergent vs. divergent thinking
  - Creativity as a skill
  - Understanding barriers: Psychological, historical and sociological
- Techniques for generating ideas
  - Idea spurring
  - Attribute listing/five sensing
  - Brainstorming
  - Parnes' "The mess"
  - Stiker's Morphology

Evaluating Alternatives

- The six factors to determining feasibility
  - Technical: is it possible?
  - Risk
  - Economic
  - Developing cost and benefit estimates
  - Legal: is it against the law?
  - Operational: will it work for us?
  - Schedule: can we deliver it in time?
- Techniques for evaluating ideas
  - Nominal group technique
  - Cost/benefit analysis
  - Delphi technique
  - Weighted-decision model

Lab Exercises: The labs for this course should include exercises where the attendee can demonstrate the ability to use critical thinking techniques and methods presented to improve their designs and approaches to development tasks.

Collaborative Troubleshooting/Problem Solving/Decision Making

Maximizing the benefits of team collective reasoning when there is a technical dilemma is a core skill for technical leadership. People tend to fall into 'hero' mode – taking on the challenge single handedly when the

situation really warrants a multi-disciplinary look, or 'hiding' mode – hoping someone else gets assigned to fix the problem or solve the dilemma. People are either afraid of blame or looking at the problem as an opportunity to shine. So how does a leader determine if the problem needs individual or team investigation? When does someone need help and how do you ensure they get the right expertise to find the problem? How does a leader establish a synergy of team effort to tackle difficult problem, focusing the energy on the best possible solution? Techniques and approaches to collaboration will be explored, practiced through role-play and discussion of scenarios.

Duration: 1 day

Learning Objectives

Upon completion of this course, attendees will be able to:

- Foster collaborative thinking and problem solving
- Establish an environment conducive to collaboration
- Effectively manage discussion in a group
- Identify barriers to effective teamwork

Course Outline

- Learn about the benefits of collaborative thinking and problem solving
- Identify ways to bring a team together to collaborate
    - Establishing the proper environment within the space and between participants
    - Communicating the time allotted, scope of effort, methods to be used
- Demonstrate understanding of ways to encourage participation
    - Encouraging idea sharing, open discussion
    - Keeping the group on topic
- Identify barriers to collaboration and ways to overcome these barriers
    - Overcoming 'group think'
    - Identifying and addressing team biases
- Maximizing the efforts of established teams
    - Getting to know each other's strengths and how to encourage those strengths

LEADERSHIP COURSE DESCRIPTIONS

Note: The instructors will be very skilled at establishing a safe, collaborative environment themselves so that the role-play and exercises are non-threatening and successful. In all courses such as this one the instructors must recognize that they are teaching technical persons these 'soft' skills that do not always come as naturally as technical skills. Participants are likely to be reluctant and may be intimidated to participate in role-play and interactive activities.

Team Building

This leadership course will be an invaluable course for personnel who are taking on a leadership role on their team. The course will provide new leaders with the core knowledge, skills, and abilities to provide direction, encourage participation, establish a 'safe' environment for sharing ideas, respect, and valuing the contributions of everyone. The course will include instruction as well as opportunity for discussion, role-play, and review of effective and ineffective leadership styles.

Duration: One Day

Learning Objectives

Upon completion of this course, attendees will be able to:

- Apply different communication strategies to maximize motivation
- Develop and maintain high levels of trust with team members
- Enhance individual and team motivation on a consistent basis
- Lead by example with competence and trust based credibility
- Eradicate issues relating to inferior quality
- Deliver higher-quality outputs more quickly and efficiently
- Deliver effective praise and reprimand sessions to reinforce standards and performance
- Create a working environment that promotes high levels of collaboration and commitment
- Set and agree on challenging performance targets with team members – and achieve them
- Effectively resolve poor performance issues within the team
- Navigate change more effectively while maintaining team focus and motivation
- Create an effective action plan to maximize motivation and performance

Course Outline

- Communication
- Developing Trust
- Enhancing Motivation
- Leading by Example
- Dealing with Issues
- Improving Efficiency
- Praise and Reprimand Sessions
- Creating an Environment for Collaboration and Commitment
- Setting Performance Targets
- Resolving Poor Performance Issues
- Navigating Change
- Creating an Action Plan

Lab Exercises: This is a highly interactive class with lab exercises that will include the opportunity for discussion, role play, and review of effective and ineffective leadership styles.